COOPERATIVE NAVIGATION BETWEEN SURFACE AND SUB-SURFACE VEHICLES

VOLUME I

JONATAN SCHARFF WILLNERS

A thesis presented for the degree of
Doctor of Philosophy

Institute of Signals, Sensors and Systems
School of Engineering & Physical Sciences
Heriot-Watt University

Submitted March, 2020

Doctoral Thesis

Cooperative Navigation Between Surface and Sub-surface Vehicles

Jonatan Scharff Willners

2019

A thesis presented for the degree of
Doctor of Philosophy

Supervised by:
Yvan R. Petillot

ABSTRACT

This thesis aims to improve autonomy and reliability in marine robots. This is achieved by combining path planning and acoustic localisation into cooperative navigation. Within path planning it is important that the robot's found path is feasible, collision-free and can be planned in real-time. This thesis reviews state-of-the-art approaches within the field of path planning along with proposing novel approaches for optimal planning under motion constraints for both goal-based and cooperative scenarios. The aim for cooperative scenarios is to support submerged vehicles with acoustic messages which can be used for localisation. This is an important problem for robots while subsurface, as it is a GPS-denied environment. Conventionally acoustic localisation has been performed by manually deploying static acoustic beacons which require extensive calibration and suffer from an operational area limited by the acoustic range of the transponders or USBL.

This thesis presents 3 major novel contributions to the field. The first one is a start-to-goal real-time path planner with path repairing capabilities for vehicles to operate in unknown environments. This extension to Hybrid-State A* makes it more useful for both planning in known and unknown environment showing a reduced computational time compared to re-planning. The second contribution presented is a leader-follower planner, where an AUV act as the leader and an ASV follows them to reduce the distance over time. This work takes the motion constraints into consideration such that it can handle all cases where the vehicle operates at different speeds, making it more generic and easier apply to multiple scenarios than classic control methods. The third major work presented is planning for cooperative missions where an ASV plans a path to position itself to reduce the navigational error on an arbitrary amount of AUVs. This work is adaptable both to the scenarios as well as the computational power on the ASV. It shows to in worst case perform as well as compared methods and in most cases outperform them by reducing the error on the AUVs with up to 60%. The results of this thesis show an improvement in the field through a combination of simulated and real data.

To my Grandparents...

Acknowledgements

I would like to give my sincerest thanks to all the wonderful people who have contributed to my time during my studies. First of all a big thanks to the Innovate UK project: Autonomous Surface and Sub-surface Survey System and to SeeByte Ltd for the funding of the position, my supervisor Yvan Petillot for the help, guidance and support. I also want to give a big thanks to USMART and ORCA-HUB for the support for trials and travels. To the people at Ocean Systems Laboratory and Seebyte, with an extra thanks to the Neptune team. I had good support from friends, and would like to extend my gratitude to Lennie, Ryan, Nicolas and Ignacio.

During my time I had the opportunity to spend 5 Months at Australian Centre for Field Robotics, where all the people were very welcoming and made the experience a wonderful one. I would like to thank all the PhD student, the marine group, and especially Lachlan for the support and Stefan for making it possible for me to come. I would not have ended up in the marine robotics field if it was not for my studies at Mälardalens University. I would like to thank the many people that I studied with, the staff and the group who participated in the development of the AUV, Naiad.

The greatest of gratitude and love to my parents, Lene and Philip and my sister Ronja for always supporting my pursuit to follow my passions (both in robotics and board sports), which by now have taken me to most continents of the world.

To my partner, Giselle, I do not think I would have managed to get through these last years as sane as I (hopefully) am if it was not for you. As for all paths, the doctoral studies involves many ups and even more downs. You have always been supportive as well as spent countless amount of hours to help me improve my work. All my love and thanks goes to you.

And in the end it all came down to the defence, which was a very pleasant experience. Thank you, Keith and Blair for taking the time to read through my work and adding insightful comments on it.

Acknowledgements - Project funding

The following companies and projects have been a part of providing funding during this thesis' doctoral studies.

SeeByte

SeeByte started as a spin-of from Ocean Systems Laboratory (OSL) in 2002 and has since then grown to become one of the world leaders in smart software for robotics with a focus on the maritime domain. The company supplies both industry and governments with their solutions.

Autonomous Surface and Sub-surface Survey System (102304)

This programme combines unmanned surface vessels (USVs), unmanned underwater vehicles (UUVs) and novel communications technology into an integrated system to provide a means of conducting low cost shore based full water column marine surveys. Success will accelerate the wider adoption of unmanned systems and will enable long term, low-cost survey and monitoring operations for offshore energy applications, deep sea mining prospecting and Carbon Capture and Storage (CCS) monitoring. There will also be a consequential reduction in the need to place humans in dangerous environments and a greater acceptability of unmanned systems by operators and regulators.

ORCA RAI-HUB (EP/R026173/1)

The Hub's primary goal is to use robotic systems and Artificial Intelligence to revolutionise Asset Integrity Management for the offshore energy sector through the provision of game-changing, remote solutions which are readily integratable with existing and future assets and sensors, and that can operate and interact safely in autonomous or semi-autonomous modes in complex and cluttered environments.

USMART (EP/P017975/1)

The aim of this project is to create a smart underwater sensing framework based on ultra-low-cost underwater communication and sensing devices ('smart dust').

ACADEMIC REGISTRY

Research Thesis Submission

Name:			
School:			
Version: *(i.e. First, Resubmission, Final)*		Degree Sought:	

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	

Submission

Submitted By *(name in capitals)*:	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals)*:			
Method of Submission *(Handed in to SSC; posted through internal/external mail)*:			
E-thesis Submitted (***mandatory for final theses***)			
Signature:		Date:	

Please note this form should be bound into the submitted thesis.

Contents

Abbreviations

ACFR	Australian Centre for Field Robotics
AD*	Anytime Dynamic A*
AHRS	Attitude and Heading Reference System
AUV	Autonomous Underwater Vehicle
ASV	Autonomous Surface Vehicle
AMV	Autonomous Maritime Vehicle
CCC	Command and Control Centre
CNA	Communication and Navigation Aid
C-Space	Configuration Space
CTD	Conductivity, Temperature and Depth
CRLB	Cramér-Rao Lower Bound
DR	Dead Reckoning
DVL	Doppler Velocity Log
D*	Dynamic A*
DP	Dynamic Programming
DW	Dynamic Window
EKF	Extended Kalman Filter
EST	Expansive Space Trees
FCL	Flexible Collision Library
FIM	Fisher Information Matrix
GIB	GPS Intelligent buoy
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GD	Gradient Descent
HA*	Hybrid-State A*

INS	Inertial Navigation System
IMU	Inertial Measurement Unit
I-RRT*	Informed RRT*
KF	Kalman Filter
LBL	Long Baseline
LPA*	Lifelong Planning A*
MC GD	Monte-Carlo Gradient Descent
MLBL	Moving Long Baseline
NLS	Nonlinear Least Squares
NTP	Network Time Protocol
MCM	Mine Countermeasures
MGD	Momentum Gradient Descent
OSL	Ocean Systems Laboratory
OWTT	One-Way-Travel-Time
PDF	Probability Density Function
PF	Particle Filter
PBHA*	Pattern-Based Hybrid-State A*
PI	Proportional Integral
PID	Proportional Integral Derivative
PRM	Probabilistic Roadmaps
RF	Radio Frequency
ROS	Robotic Operating System
ROV	Remotely Operated Vehicle
RRT	Rapid-exploring Random Trees
RRT*	Asymptotic Optimal RRT
TDMA	Time-Division Multiple Access
TDoA	Time Difference of Arrival

ToF	Time of Flight
ToL	Time of Launch
TWTT	Two-Way-Travel-Time
SBL	Short Baseline
SGD	Stochastic Gradient Descent
SLAM	Simultaneous Localisation And Mapping
USBL	Ultra-short Baseline
UKF	Unscented Kalman Filter

List of Figures

List of Tables

List of Algorithms

Publications

The work that lead up to this thesis have contributed to the following publications.

[OCEANS'17] **J. S. Willners**, P. Patron, and Y. R. Petillot, *"Moving Baseline Localization for Multi-Vehicle Maritime Operations"*, IEEE OCEANS, (Aberdeen, Scotland), June 2017.

[OCEANS'18] **J. S. Willners**, Y. R. Petillot, P. Patron and Daniel Gonzalez-Adell, *"Kinodynamic Path Planning for Following and Tracking Vehicles"*, IEEE OCEANS, (Charleston, USA), October 2018.

[OCEANS'19] **J. S. Willners**, L. Toohey and Y. R. Petillot, *"Reducing Uncertainty From Range-Only Localisation by Transmission at Optimal Time"*, IEEE OCEANS, (Marsielle, France), June 2019

[RA-Letter/IROS'19] **J. S. Willners**, L. Toohey and Y. R. Petillot, *"Sampling-Based Path Planning for Cooperative Autonomous Maritime Vehicles to Reduce Uncertainty in Range-Only Localisation"*, IEEE RA-Letter/IROS, Macau, November 2019.

[AUV'18] D. A. Robb, **J. S. Willners**, N. Valeyrie, F. J. C. Garcia, A. Laskov, X. Liuy, P. Patron, H. Hastie and Y. Petillot, *"A Natural Language Interface and Relayed Acoustic Communications for Improved Command and Control of AUVs"*, IEEE OES Autonomous Underwater Vehicle Symposium, (Porto, Portugal), November 2018.

[DIS'19] David A. Robb, J. Lopes, S. Padilla, A. Laskov, F. Garcia, X. Liu, **Jonatan Scharff Willners**, N. Valeyrie, K. Lohan, D. Lane, P. Patron, Y. Petillot, M. Chantler, H. Hastie, *"Exploring Interaction with Remote Autonomous Systems using Conversational Agents"*, The 2019 ACM conference on Designing Interactive Systems, (San Diego, USA), June 2019.

1 Introduction

"The surface of the Earth is the shore of the cosmic ocean. From it we have learned most of what we know. Recently, we have waded a little out to sea, enough to dampen our toes or, at most, wet our ankles. The water seems inviting."

- Carl Sagan

The ocean is the origin of all life we know of in our universe. It is the first thing we look at in the search for extraterrestrial life. This source of life is also the main contributor to the air we breath, with the phytoplankton in it generating up to 85% of the oxygen in our atmosphere. It controls the weather and the climate on earth. In 1890, Alfred Thayer Mahan published the book *The Influence of Sea Power Upon History*, where he proclaims that to control the sea is the key to a nations power, both commercially and military. Despite immense technological advancements since the late 19th century, this statement still holds true. The ocean is still the main medium for transportation of goods with over 90% of trade between countries performed by ships. It is by all means the most important resource on earth, as much to the people living in a remote village as it is to whole nations. Yet, we have explored just a fraction of it. As today, Over 90% of the ocean remains unmapped, while our closest astronomical bodies, the moon and Mars, have accessible and detailed maps of them. This thesis will provide suggestions on how to improve the mapping of oceans by improved autonomy and reduced position error to improve the georeferencing of collected data.

1.1 Motivation

The ocean is an unpredictable, dangerous and unforgiving environment. There are high risks associated with manned underwater work, both in vehicles and by divers. Robots have the possibility to replace, or work cooperatively with humans to achieve the same goals. They can perform long-term missions (up to months at a time) and operate in the deep ocean. Missions for maritime environment could include using a Remotely Operated Vehicle (ROV) to inspect and repair infrastructure or Autonomous Maritime Vehicles (AMVs) to gather environmental data. The difference between ROVs and AMVs (which includes Autonomous Underwater Vehicles (AUVs) and Autonomous Surface Vehicles (ASVs)) is based on the level of autonomy. An example of the different types of vehicles can be seen in Fig. 1. An ROV is tethered and controlled by an operator whilst AMVs operates autonomously to solve a task or objective. As ROVs require tethered operations, this is usually performed from large ships, including the crew and the ROV-pilot. The use of ships with staff is an expensive operation. The autonomy of AMVs can significantly reduce the cost of such operations by removing the need for a ship and a lot of the personal required. Small AMVs such as number 1 and 2 in Fig. 1 can be launched from shore or by a small boat by as little as one or two people while larger vehicles and ROVs often need a boat along with a crew.

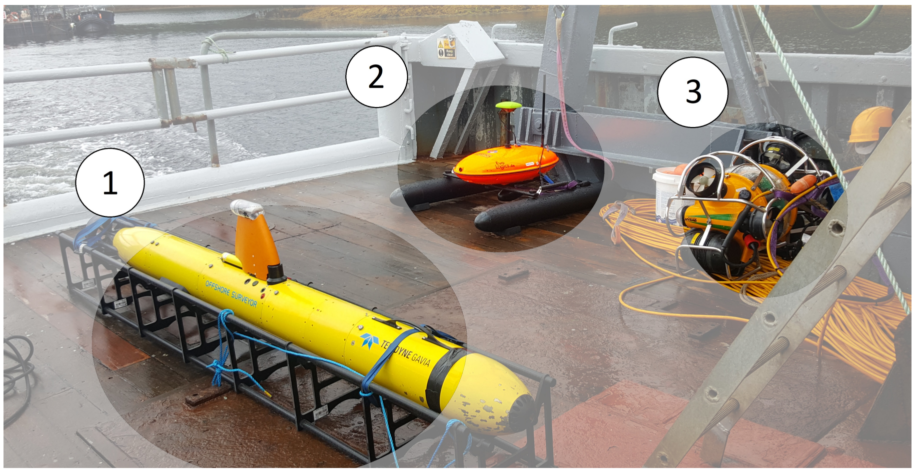


Fig. 1: Different types of unmanned vehicles are categorised based on their level of autonomy. (1) AUVs and (2) ASVs are un-tethered and work autonomously to solve objectives. (3) ROVs are tethered and require continuous human interaction.

There are many benefits from using AMV. This has caught the attention of various scientific areas and industries to develop and use marine robots in their field of work. A selection of these include:

- **Defence:** Out of the earth's 197 countries, 146 have a coastline. Maintaining control over coastlines is a key element to protecting the country [1]. Maritime robots are being used in the defence sector in jobs ranging from ensuring safe passages for ships to searching, classifying [2] and disposal of mines (Mine Countermeasures).
- **Industry:** The oil and gas industry use marine robots to repair, inspect and maintain infrastructure [3, 4]. The usage of commercially available platforms grows each year; the predicted growth between 2018 and 2022 is 74% [5].
- **Oceanography:** Marine robots help collect data used for environmental monitoring [6], coral reef conservation[7], deep-sea and arctic exploration [8]. Data from such missions can help scientists understand how life emerged, forecast the weather and how the climate is changing.

- **Archaeology:** The work of exploring, locating and observing sunken ships [9] or structures [10] have conventionally been performed by divers. As AMVs are becoming more accessible, this field has expressed growing interest in using autonomous vehicles that can perform these missions for longer time, at deeper scenes and with a higher precision than divers.

- **Search and rescue:** AMVs have been used in search and rescue missions from accidents as plane crashes [11] and natural disasters such as hurricanes [12, 13] as they can help cover large grounds without much human interaction and operating close to 24 hours a day.

A shared problem for marine vehicles is that water highly attenuate high frequency radio signals. As a result there is no access to Global Navigation Satellite System (GNSS) (and hence no Global Positioning System (GPS)). For un-tethered vehicles communication is also highly limited in both range and bandwidth, as they have to rely on acoustics communication while submerged. Without access to GPS while submerged, the vehicles have to rely on other methods to estimate their position. This is most commonly performed by Dead Reckoning (DR), which is a method to integrate sensor data over time. Due to noise and drift in sensor, DR results in having an error that grows without bound over time. The rate at which the error grows is dependent on the quality of the sensors available. To achieve a low drift DR ($<1\%$ error of the distance travelled [14]) the vehicles need to be equipped with an expensive and high precision Doppler Velocity Log (DVL). If a vehicle however is not equipped with a DVL the error can reach above 20% of distance travelled. The heading of the vehicles can also be drifting or have an offset. This causes an issue related to the quality of collected data. If a vehicle cannot georeference an observed feature with high enough reliability for re-acquisition in a subsequent mission, the data loses much of its value. This data is often collected by sonars or cameras, which can also be used for Simultaneous Localisation And Mapping (SLAM). SLAM is one method that can sense the natural environment and use this as a source for localisation[15, 16, 17, 18]. However when this methods rely on observation and (re-observation) of distinguishable features in the environment. For optical sensors this is not always possible due to adequate water conditions. Another method to bound the positional error from DR is by using acoustics localisation. Acoustics can be used to measure the angle or distance of an incoming acoustic transmission. Using the angle or distance along with knowledge about where the signal originated from, acoustic localisation methods can be applied [19, 20, 21, 22, 23]. These methods are however dependent on additional infrastructure or vehicles, but is not reliant on re-observing features in the environment and can hence be used in most locations. Localisation using acoustic communication can be done in multiple ways. The traditional approach is Long Baseline (LBL) which is based on using multiple wide-spread, pre-deployed, static and calibrated acoustic beacons. ASVs have recently been used to replace these beacons. This overcomes many of the shortcomings associated with LBL beacons. Using mobile beacons enables the beacons to adapt to the AUVs and they can position themselves in strategic beneficial positions to improve the acoustic localisation technique. An other approach using acoustic communication is Ultra-short Baseline (USBL), where the direction of an incoming transmission is measured. If this direction of the signal is used in addition to the range a position can be estimated. These systems are often easy to deploy but more expensive [24] and the direction measurement includes noise, which over long distances can translate to large error, it does however bound the error.

1.2 Problem Statement

The use of robotics for maritime missions includes mine countermeasures [2, 25], bathymetric surveys and search- and rescue-missions. These missions share the need to be able to geotag collected data. The quality of the data gathered from such missions is related to how well it can be geotagged. An object of potential interest loses much of its value if the uncertainty in its position is too high to enable re-acquisition in a subsequent mission [25, 26, 27]. If a potential target of interest cannot be re-observed in a subsequent mission due to low localisation accuracy, a new survey might be required to rediscover the target. This is a time consuming and expensive task. Other risks associated with unreliable localisation is increased risk of collisions or loss of vehicle. In multi-vehicle operations where a vehicle (from here on referred to as a Communication and Navigation Aid (CNA)) is dedicated to support others with acoustic localisation messages can reduce this error, making it a safer and more reliable operation for AUVs. During such multi-vehicle missions, it is also important that the CNA operates in a safe manner. For a vehicle to move in a safe manner it is important to ensure that its path is feasible and collision free. Planning a path for a CNA to improve the localisation on AUVs can be seen as a combination of the fields; path planning and acoustic localisation. It involves finding where to transmit a localisation message from, such that the effect is as great as possible. The position to achieve this is based on the localisation method used and the geometrical relationship between transmitter and receiver [28].

1.2.1 Geometry's Effect on Acoustic Localisation

The geometry between acoustic beacons and an AUV affect the result of the localisation method [28, 29], such as the distribution of the beacons around the AUV. Other factors which affect the result are based on noises in the measurements. This can be a product of erroneous estimation of the sound velocity in water [30]. The estimation of the water channel is based on multiple factors, some of which are measurable, Conductivity, Temperature and Depth (CTD)[31], and some which are harder to measure such as multi-path. The ones related to the water channel are hard to control, however the geometrical relationship between vehicles can be influenced by moving the beacons for transmission at positions which are calculated to have a better affect.

1.2.2 Path-Planning for Autonomous Maritime Vehicle

Ensuring safe path planning for AUVs and ASVs is critical. If collisions occurs at sea, this might lead to leaks which with high probability can cause a loss of the vehicle or cause damage to the embedded electrical systems. Collision free path planning needs to take the motion constraints of the vehicle into consideration. The paths also need to be planned in real-time, or close to, as many vehicles are under-actuated and need to be in constant motion to maintain control, and can therefore not be idle and wait for a long time for a planning procedure to finish before executing the path.

1.2.3 Cooperative Navigation

Cooperative navigation in this thesis refers to how a vehicle can adapt its plan based on others. This can be to follow a vehicle, or to position itself to improve the acoustic localisation. Such plans need to be planned in real-time and take motion-constraints into consideration while finding paths that incorporate knowledge about how the geometrical configurations between the CNA and the AUVs affects the acoustic localisation methods used.

1.3 Objective of the Thesis

This thesis explores methods on how to solve some of the path planning problems related to autonomous operations in the maritime environment. While the presented algorithms developed through the work that lead up to this thesis are path/motion planners, it is heavily influenced by the effect the geometric relationship between the platforms has on the localisation of the submerged vehicle. This thesis is therefore a combination of two fields; acoustic localisation and path planning, the thesis objectives are as follows:

- **Review of acoustic-localisation methods:** To perform a review and comparison of the most used range-only localisation methods and filters. As one of the main goals of this thesis is to use ASVs as navigational beacons, to understand range-only localisation is essential to establish its potential and limitations. With these vehicle's ability to move, they can position themselves in strategic positions which can achieve a better configuration to improve acoustic localisation.
- **Path Planning for Maritime Vehicles:** To develop new methods for path planning it is important to review the field. From this, suggestions on how to achieve reliable, collision-free path planning methods will be developed.
- **AUV-ASV Cooperative and Adaptive Planning:** To contribute to the field, novel suggestions for how to plan vehicles in an adaptive and cooperative approach are suggested.
- **AUV-ASV Cooperative Localisation:** A review of the field and suggestions for a novel path planner for ASVs for cooperative localisation algorithm are proposed.

This algorithm helps increase the performance of the range-only localisation algorithms.

1.4 Novel Contributions

This section will outline the novelty that this thesis has brought to the field.

1.4.1 Start-to-goal Path Planning

The thesis presents a novel extension for state-to-goal path planning in chapter 4. In this chapter an extension to Hybrid-State A* (HA*) is presented to handle real-time planning and path-repair using previously explored paths of the search tree. The benefit of this is that when a vehicle is operating in an unknown environment and senses an obstacle which obstructs the current plan it can repair the path by pruning the tree from non-feasible branches and use the pruned tree as a start of a new search. This reduces the search time making it suitable for real-time exploration of unknown environments.

1.4.2 Leader-follower

An planning algorithm to solve the leader-follower problem from the followers perspective is presented in chapter 5. Compared to traditional control techniques this approach can handle scenarios where the motion capabilities of the vehicles differ such as the follower being a non-holonomic vehicle with a minimum speed greater than the leader's maximum speed and the other way around as well as when they can operate at the same speed. The same algorithm can also solves multiple of other scenarios such as having a vehicle loitering around a specific point or following an inspection vehicle to support it with navigational data. The algorithm can be used on arbitrary vehicles for multiple dimensions with small modifications — this thesis shows it operating in 2D and 3D.

1.4.3 Cooperative Navigation

A spatio-temporal planner is presented in chapter 6 which can improve the positional accuracy for acoustic localisation by positing a mobile transponder at waypoints with geometrical advantages for the receiving vehicle(s). The algorithm can in theory support any number of vehicles. The novelty for this one is how it combines planning using a priority-based search tree with stochastic expansion of the tree and a temporal optimiser to find both from where at when an acoustic message should be transmitted. The algorithm is adaptable to the computational power of the vehicle using it by changing 2 parameters that represent how it should focus on finding closer to optimal path and how much it should focus on exploration. A sub-problem of this related to selection of when to transmit acoustic messages is presented in section 5.4 which is a small framework which can reduce the acoustic localisation error without interfering with any objectives the vehicle in question is performing.

1.5 Thesis Structure

The thesis will first perform a review of the two separate fields which it builds on: AUV localisation using acoustics and path planning. The work after this follows a trajectory beginning at planning feasible path to reach a certain position to planning a feasible path to follow a target. This is extended to select where along a planned path it is most beneficial to transmit localisation messages from such that a receiving AUV's positional error is estimated to be reduced the most. The last technical chapter presents a method to find a path for a vehicle, to decide when space and time to transmit localisation messages would be most beneficial for receiving AUVs. How the different chapters relate to each other can be seen in Fig. 2.

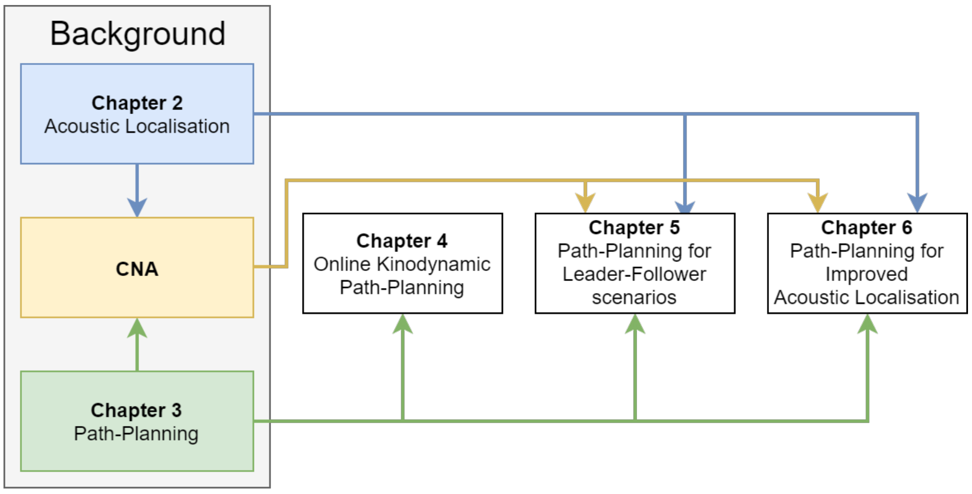

Background
Chapter 2
Acoustic Localisation
CNA
Chapter 3
Path-Planning
Chapter 4
Online Kinodynamic Path-Planning
Chapter 5
Path-Planning for Leader-Follower scenarios
Chapter 6
Path-Planning for Improved Acoustic Localisation

Fig. 2: The thesis chapters and how they relate to each other.

Chapter 2. [*AUV Localisation Using Acoustic Communication*] Pages: 11 - 51. This chapter introduces the issue of not having access to GPS underwater, and how using acoustic communication can be implemented as an alternative method of localisation. It describes the various configurations and some of the more commonly used methods used for distance based localisation and how they are affected by different geometrical relationships.

Chapter 3. [*Path planning*] Pages: 52 - 72. This chapter presents some of the most used path planning algorithms along with some extensions of them. It presents some work on real-time planning along with how to ensure collision free paths under motion constraints.

Chapter 4. [*Online Kinodynamic Path Planning and Re-planning in 3D*] Pages: 73 - 90.
This chapter presents a path planning framework with the goal to reach a goal region based on guided expansion of a search tree. It takes the motion constraints of the vehicle as well as environmental limitations into consideration.

Chapter 5. [*Vehicle Tracking and Following under Kinodynamic Constraints*] Pages: 91 - 110.
This chapter presents a path planner for a *follower* to autonomously track and follow a *leader*. The aim is to plan a feasible path under motion constraints which reduces the average distance between the leader and follower vehicle over time. It was developed to especially handle cases where the follower has a minimum speed which is greater than the leader's maximum speed.

Chapter 6. [*Spatio-temporal Path Planning to Improve Range-only Localisationn*] Pages: 111 - 126.
A spatio-temporal path planner for a support vehicle to find transmission positions, along with times to transmit acoustic localisation messages from to reduce the uncertainty on an arbitrary amount of submerged vehicles.

Chapter 7. [Conclusion] Pages: 127 - 130.
This chapter summarises the thesis, reviews the contributions and suggests future work which is either planned or could be continued from this thesis.

1.6 Mathematical Notations

The following mathematical notations will be used throughout the remainder of the thesis.

$\|\|A - B\|\|$	$\triangleq$	Euclidean Distance between A and B
$\|A\|$	$\triangleq$	Absolute value of A
$\mathbb{R}$	$\triangleq$	Real set of numbers
$\mathbb{R}^n$	$\triangleq$	The real set of numbers in n dimensions
v_{sound}	$\triangleq$	The speed of sound in water
$\mathcal{N}(\mu, \sigma)$	$\triangleq$	Gaussian Distribution with mean μ and variance σ
$\mathcal{C}$	$\triangleq$	Configuration Space
$\mathcal{C}_{free}$	$\triangleq$	Free Configuration Space
q	$\triangleq$	Configuration i in Configuration Space
φ, θ, ψ	$\triangleq$	roll, pitch, yaw

Tab. 1: Mathematical Notations

2 AUV Localisation Using Acoustic Communication

"Not all those who wander are lost"

- J.R.R Tolkien

Every year AUVs see increasingly more usage. They are gaining higher autonomy and are collaborating to complete objectives or to map areas in less time [11]. One area that is unsolved, and will be until new methods are developed, is absolute localisation. Vehicles with access to GPS have a reliable source to estimate their position within a few meters. However in GPS-denied environments such as inside buildings, caves and subsurface this is not an option. For subsurface vehicles this is due to the characteristics of water, which attenuates high frequency signals such as Radio Frequency (RF). The lack of certainty in the vehicle's position brings additional issues regarding georeferencing collected data with precision. The quality of gathered data loses much of its value if the uncertainty in its position is too high to enable re-acquisition in a subsequent mission. Too high uncertainty in a vehicle's position estimate also increases the risk of collisions, redundant data collection or in the worst case scenario loss of assets. As these are clearly all undesirable events, the need for reliable localisation is important. On the surface the vehicles have access to GPS, but as soon as they dive they have to rely on DR until the next update from an external source. DR is performed by integrating sensor data over time. This is commonly based on data from a set of navigational sensors which could include DVL, Inertial Navigation System (INS), Inertial Measurement Unit (IMU), compass and/or Attitude and Heading Reference System (AHRS). DVL is used to estimate the sensors velocity in relation to eg. the bottom using acoustics. The DVL senses velocity in the local X,Y and Z coordinate frame. By using fusing this data with the data from an IMU and/or compass to measure the heading of the vehicle an estimation of the vehicle's position can be made. Data from sensors contain noise, and as DR performs the integration of this, the state estimation will include all errors and noise since the last absolute position measurement. The error from DR can usually vary from 1% using expensive modern equipment to over 20% when inexpensive and limited sensors are used in combination with motion estimates. For shorter missions the error produced (from vehicles equipped with sensors resulting in errors in the lower range) might be acceptable. For longer missions an option is for the vehicle to surface to gain access to GPS during the mission/objective. However, every time the vehicle surfaces, the action takes valuable energy and time from the main objective(s). For certain missions such as objectives in deep water, this may not even be a viable option as the error would have grown too large once the vehicle has reached its operating depth again to achieve bottom lock with DVL. It is easy to understand that DR is far from a long-term solution for all types of missions. Instead more information is needed to be used jointly with DR. By combining DR with external sources of information, the otherwise continuous growth of error could be bounded. One potential approach is SLAM [32]. SLAM continuously builds a map of its surrounding and uses the map to localise itself within. It has seen some usage for AUVs [15, 16, 33, 34]. For a vehicle to localise itself within a map, SLAM is reliant on detecting and re-observing features. Water is however an environment with many limiting factors for many sensors to be used properly, including turbidity and low brightness. These factors can limit the possibility to extract enough features to reliably match them to the ones previously observed in the map. With too few re-observed features it is hard to make an accurate state estimate in SLAM. While there are feature-less approaches for SLAM [35] they still wont supply a solution in too unclear water. Hence using vision [15] laser or sonar [36] based approaches for SLAM is too unreliable to be considered an absolute navigational solution in all conditions. This is however not the choice of using either SLAM or acoustic localisation as they are more commonly used together. Other methods have been examined such as terrain-aided-navigation[37], but fail to achieve the precision in localisation which is needed to be able to re-visit a place of interest but could however be good enough for other scenarios such as mapping of temperature. The precision needed to re-visit an object of interest differs depending on the objective and the sensor. If side-scan sonars are used, the precision need is lower than the usage of optical sensors, where the vehicle need to be up close to get a good image. Another option is to use acoustics as an external reference. This can be performed by using the distance or the angle towards a source with a known position. These methods of estimating the vehicle's position fall into the categories of trilateration or triangulation. Trilateration is based on estimating a position based on distance to known positions while triangulation uses angles. These are well used approaches and have been proven useful since the 1960's. These approaches show some benefits over SLAM as they are not reliant on re-observable features nor affected by turbidity and light. However, they have drawbacks including a reliance on additional infrastructure (such as multiple vehicles or deployed transponders) and an operational region that is limited to the range of the acoustic signals from the transponder(s). Nonetheless, acoustic localisation can be seen as a more general approach for absolute localisation. It can be used to localise a vehicle which is completely lost, where as relying on a map constructed by the vehicle can be erroneous nor is it dependent on adequate water conditions. In this section we will present some of the more common approaches on how to use acoustics for localisation. It will explore how to improve the localisation on AUVs from the transmitters' point of view. That is, how can a transmitter be placed in relation to the receiver(s) to reduce the error to the greatest extent.

This chapter will in section 2.2 give a brief introduction to underwater acoustic localisation, including its advantages and shortcomings. Section 2.1-2.4 will give an introduction to the problem and the necessary components to solve it. In sections 2.5 - 2.7 different localisation methods and filters using acoustics are presented. Section 2.8 will present a comparison of described localisation methods. Section 2.9 presents the concept of CNA and 2.10 will summarise this chapter.

2.1 Problem Formulation

By using acoustic localisation, the otherwise continuous growing error from DR can be bound. In this thesis we will consider this from a distance-based approach. That is, how to estimate a position based on distances to known positions. To do so we need to know how to do the the following:

- How to obtain distance measurement from acoustic communication. This will be discussed in section 2.3.1.
- How to estimate a position based on internal sensor data (DR). This will be discussed in section 2.4.

To achieve a method of global positioning we need to combine these to solve the non-linear problem of finding the intersection of circles/spheres based on distance to their centres. This can be divided into two approaches to obtain a position estimate; the first one is that we try to solve the trilateration problem. The second one is how to use filters to get a Bayesian estimate of the position based on the knowledge of the system.

2.1.1 Trilateration

The trilateration problem is to find a position where multiple circles/spheres intersect. This is, if we have multiple distance measurements we try to find the AUV's estimated position $\hat{X}$ based on distance measurements $Z_{1,..,i}$ to the transmission positions $P_{1,..,i}$. Based on this, we want to find the position that is most likely to be the real position of the robot. To do this we want to find the point closest to the intersection between the distance measurements to the known positions such as:

$$\hat{X} = \underset{X \in \mathbb{R}^n}{\arg\min}\, E(X, P, Z) \tag{1}$$

How equation (1) can be solved will be shown in section 2.5.

2.1.2 Bayesian Estimation

Another approach is to use Bayesian estimation, to use a filter that updates the position based on the most likely belief based on the previous state and the measurements. This can be generalised as estimating the probability of the state x at the current time as a function of the measurements z over time as:

$$p(x_t|z_{1:t}) \tag{2}$$

This is commonly solved by recursive filters. Two such ones will be presented in the following section to solve the localisation problem:

- Extended Kalman Filter, an extension to the Kalman Filter to handle non-linear measurements (such as distance based). This filter will be described in section 2.6.
- Particle Filter, a Monte Carlo approach which can handle non-linear measurements. This filter will be described in section 2.7.

2.2 Underwater communication

To be able to communicate with an un-tethered submerged AUV, wireless communication is needed. There are three methods that can be used for this, acoustics, optical and RF. The three different mediums are affected differently in water. This section will give some of the main issues with each, along with other characteristics of communication in water.

RF is on land the most common way of wireless communication. As it is able to use high frequencies, high bandwidth can be achieved. However, this is a disadvantage when used underwater as the signal loss in water grows with the frequency, hence, high bandwidth RF signals are only usable for very short ranges ($<$ 10 metres) [38]. To transmit over longer ranges (few kilometres), low frequencies are needed (3-30kHz have been used) which have a low bandwidth and require large equipment and high power [38].

Optical communication is another method to communicate with high bandwidth. Water has a lower absorption at the wave length of ultra-violet and the visible spectrum for humans as can be seen in Fig. 3. This makes it an alternative approach for high-bandwidth communication underwater. However, it is dependent on line of sight, something that is highly affected by the water quality. Another issue is the light from the sun, which creates light waves in the same spectrum as the optical communication and hence will add noise. If there is too much external light, optical communication might not be usable. This can limit the method to not being able to be used while the sun is shining.

The last method of communication is acoustics. Compared to the previous methods it consumes a lot of power in regards to the amount of data transmitted as it needs to create

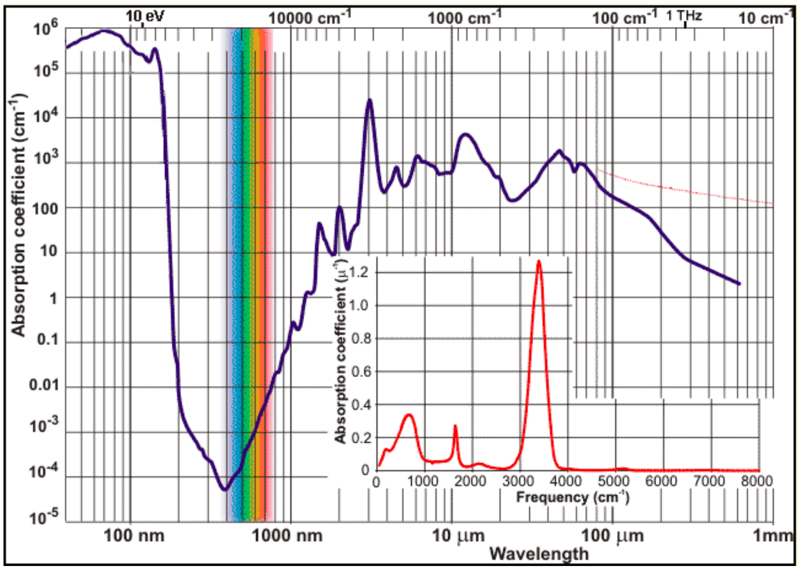


Fig. 3: There is a spectrum for light with lower absorption coefficient which enables the usage of optical communication underwater. Image-credit [39].

mechanical waves. However, it is low frequency (<300Khz, commonly 12k-48kHz in commercial systems) which means it is not absorbed as much as the other methods making it able to be used over greater distances (10's of kilometres with powerful transducers, <10 kilometres with commercial systems for AUVs).

Based on the drawbacks of RF and optical communication, acoustic is the dominant means of communication with submerged platforms. The propagation of acoustic signals has increasing attenuation with frequency [30], hence long range communication needs to be low bandwidth. Acoustic signals are heavily affected by multi-path propagation [40] and has a low propagation speed which varies with the acoustic velocity profile, which CTD sensors can help to estimate [41]. How the acoustic velocity is affected by the depth can be seen in Fig. 5. The typical properties and characteristics for acoustic communication are described below.

- **Low bandwidth**
  The path loss of a signal in water is dependent on the frequency and the distance travelled [30]. The strength of the signal decreases by distance due to the spherical spread of the acoustics and the attenuation. And the higher the frequency, the higher the attenuation in water. Hence, to transmit for long range either much energy is needed an/or low bandwidth is required.

- **Multipath** Multipath is caused by signal reflection (from bottom, surface or an object) and refraction (variances of the propagation speed within the water, see Fig. 5). An example of the various paths the acoustic signal can take can be seen in Fig. 4. This can cause the same signal to arrive at multiple times, which can lead to the information in the signal being corrupted by collisions.

- **Low Propagation Speed - High Latency:**

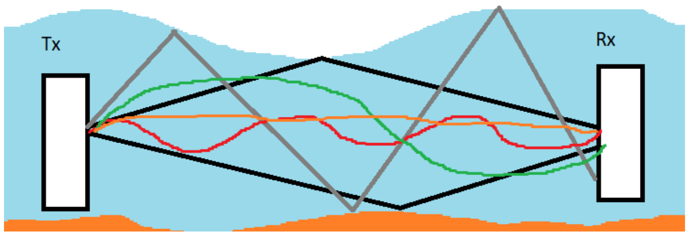


Fig. 4: Multipath is an issue for acoustic signals. The same signal can be received at multiple times due to different paths.

Speed of sound in water is of many magnitudes lower than the speed of electromagnetic waves. As such the delay or latency in communication can be in the order of seconds over long distances. The speed varies with CTD as can be seen in Fig. 5. From [31], equation (3) is used to calculate the speed of sound in water where T is the Temperature in centigrade, C Conductivity which is based on the salinity (PPT) and D is the Depth in metres.

$$V_{sound} = 1449.2 + 4.6*T - 0.055*T^2 + 0.00029*T^3 + (1.34 - 0.01*T)(C - 35) + 0.016*D \tag{3}$$

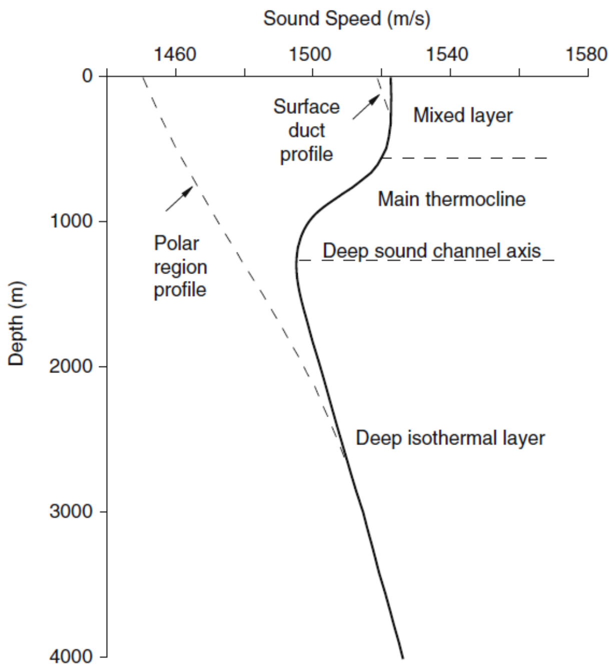


Fig. 5: Speed of sound in water is dependent on water density and compressibility (Conductivity, Temperature and Depth). Image-credit [31]

- **Thermocline layer:**
  In the oceans there is a certain depth where the sound velocity of acoustic signals

rapidly changes. This is called the thermocline layer and will create a shadow zone where the acoustic signals cannot enter due to the layer acting as a reflector, which is a part of the multipath problem. Submarines during covert missions on the other hand, use this to their advantage to remain unnoticed from sensors above the thermocline layer. For ASV-AUV missions, it does however remove or reduce the possibility of how to communicate with deep operation AUVs. A representation of the effect can be seen in Fig. 6.

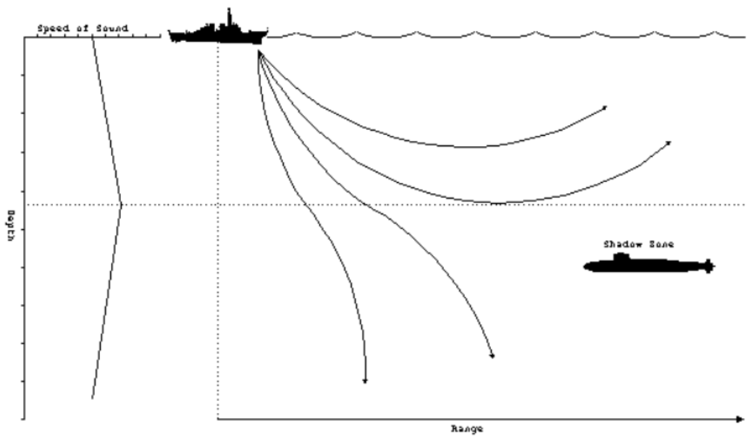


Fig. 6: The negative sound speed gradient of the thermocline layer creates a shadow zone below it, making it *impossible* to reach with acoustic signals in certain areas. Image-credit [42]

- **Noise:**
  Noise comes in many forms in the ocean including turbulence, surface motions and thermal as well as from the vehicle it self. Surface motion (rain, other vehicles, wind causing waves etc), is the major source of noise for the frequencies 100 Hz - 100 kHz (which are the operating frequencies of most acoustic modems).

As stated, the longer the distance to transmit, the lower the data rate and higher the energy is needed. Transmitting shorter distances with multi-hop and at higher data rate can in many cases be both faster and require lower energy consumption [40].

2.3 Localisation using Acoustic Communication

As mentioned, DR is a method with continuously growing error and uncertainty. This error can be bounded by using external references. One type of external reference can be obtained from using acoustic signals/communication with transponders with known locations. In the early stages of acoustic localisation for underwater vehicles, Long Baseline (LBL) was the primary method. LBL is based on measuring distances or Time Difference of Arrival[43] to widely spread transponders, as seen in Fig. 7.

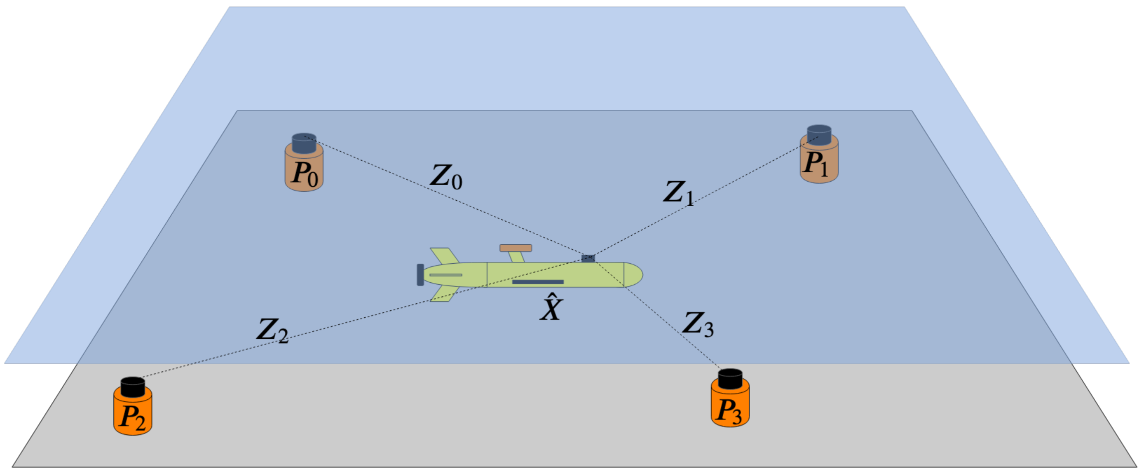


Fig. 7: Long Baseline can be used to measure the distance to multiple transponders with known locations. This can be used to calculate the AUV's position.

LBL systems have been used since the 1960's and are still in use today. It has achieved sub-centimetre precision in a small confined environment (7.75m diameter, 4.25m deep tank) using a high-frequency (300kHz) LBL system [44] which achieved an update rate of 5Hz. However, due to the rapid attenuation of high-frequencies in water, this is not applicable to many open sea operations or for larger surveys. Instead lower frequencies (12-48kHz) have been used to allow longer range [45]. The trade-off for using lower frequencies to achieve a longer range is less precision and longer period between updates. A 12kHz system presented by Hunt *et al.* [45] achieves a precision of 10m at a 2km range every 20 seconds. LBL systems are conventionally moored to the sea-floor, making it a time consuming task to deploy and recover the transponders. As the transponders are in a static position underwater they need to be localised before they are able to be used as a source for localisation [46, 47]. This means that time is also needed for calibration of the system.

An alternative approach to moored transponders for LBL is to use GPS Intelligent buoys (GIBs). Using buoys with GPS access reduces the time for deployment and removes the need for localisation of the transponders before usage [48, 49].

Another method is the usage of USBL or Short Baseline (SBL), which consists of multiple hydrophones to measure the Time Difference of Arrival (TDoA) and phase shift in the signal between the hydrophones to calculate the slant and bearing angle of the acoustic signal [50]. The response from the USBL equipped platform as an acoustic message can then help the AUV to localise itself by combining the USBL measurement with DR [51, 52]. Cooperative navigation is a field that is getting more attention recently which makes use of the concept of Moving Long Baseline (MLBL), which instead of moored beacons or GIBs uses multiple vehicles instead. This approach brings multiple benefits such as ease of deployment and eliminates the need to manually deploy, localise and retrieve static beacons. The operating areas for the AUVs are no longer limited by the communication range of the static beacons, as the mobile ones can follow the AUVs [20]. An alternative to using multiple transponders to perform LBL either though static or moving transponders is single beacon localisation. This method type of configuration either uses delayed state forms for localisation methods or filters. It reduces the number of transponders needed for localisation, but at the cost of potentially lower precision due to a shorter *baseline* between transmissions. A summary of some of the benefits and drawbacks of various configuration can be found in table 2.

Method	Pros	Cons	Examples
LBL (Moored)	High precision	Limited operational region, Time consuming to deploy, retrieve and calibrate	[53, 54]
LBL (GIB)	Faster deployment and no calibration needed	Limited operational region, manual deployment	[49, 55]
MLBL	Mobile	Requires multiple vehicles	[20, 21, 56]
Single beacon	Cheaper, can be mobile	Lower precision	[22, 23, 57, 58]
USBL/SBL	Can be mobile	Synchronisation with position and orientation of vehicle equipped with the USBL/SBL, expensive, not as high precision as LBL	[59, 60]

Tab. 2: Benefits and drawbacks from different transponder configurations. The benefit of having a mobile beacon is that it overcomes the shortcoming in which the operational area is the region to where the transponders are deployed.

2.3.1 Distance measurement

Using the distance acoustic signals(s) have travelled and the origin of the signal(s) can be used to localise a robot. To do this we need to be able to measure the distance. This can be done in two ways; One-Way-Travel-Time and Two-Way-Travel-Time

2.3.1.1 Two-Way-Travel-Time

The easiest form to measure distance between a robot and another beacon/transponder is by using an exchange of messages. This is called Two-Way-Travel-Time (TWTT). It is based on transmitting a signal at a known time, and waiting for a response as seen in Fig. 8.

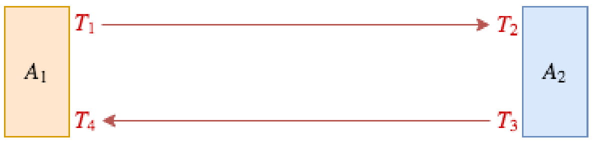


Fig. 8: A message exchange between platform A_1 and A_2 can be used to synchronise A_1's clock in respect to A_2 or for A_1 to measure distance between them.

By using the time-stamps ($T_{1,2,3,4}$) seen in Fig. 8, equation (4) can be used to calculate the round-trip time of the message. This time can then be used with a the sound velocity profile to calculate a range as in equation (5).

$$t = (T_4 - T_1) - (T_3 - T_2) \quad (4) \qquad d_{TWTT}(t) = \frac{t}{2} * v_{sound} \quad (5)$$

2.3.1.2 One-Way-Travel-Time

The other method to measure time is by using a single message instead of an exchange. This is referred to as One-Way-Travel-Time (OWTT). Instead of using all the time-stamps ($T_{1,2,3,4}$) as seen in Fig. 8, only the ones from one message are needed. If vehicle A_2 receives a message at time T_2, transmitted by vehicle A_1 at time T_1, equation (6) can be used. However, there is one pre-condition for this equation; A_2 needs to be able to convert T_1 to A_2's time-frame. This means that the vehicles' transmitter/receiver need to be synchronised in time [61].

$$d_{OWTT} = (T_2 - T_1) * v_s \quad (6)$$

The time-synchronisation process can either be performed on the surface by (as an example) using the time from GPS or submerged using a synchronisation protocol. This could be performed by Network Time Protocol (NTP). Performing a time-synchronisation requires a message exchange as in Fig. 8. To calculate the offset of A_2 in relation to A_1 equation (7) can be used.

$$Offset = \frac{(T_2 - T_1) + (T_3 - T_4)}{2} \tag{7}$$

A drawback of OWTT is the requirement of synchronised clocks as well as the need to periodically re-synchronise the clocks due to drift (although atomic clocks can provide an accurate time-synchronisation over a long time [62, 63]). An additional benefit of having synchronised clocks is that one vehicle can support an arbitrary number of vehicles with a single message, instead of a message exchange with each vehicle. As the vehicles do not need to exchange message to estimate the range between them, it can be used as a silent positing method for the duration the clocks are synchronised. Another silent positioning approach is TDoA which measures uses the time difference of multiple received signals from know origins. However, it relies on either multiple transponders or the receiver being in a static position.

2.3.1.3 TWTT-OWTT Comparison

A comparison between OWTT and TWTT, as in Fig. 9, shows a slight offset between the methods which can be caused by clock-drift and various filter- and hardware-delays.

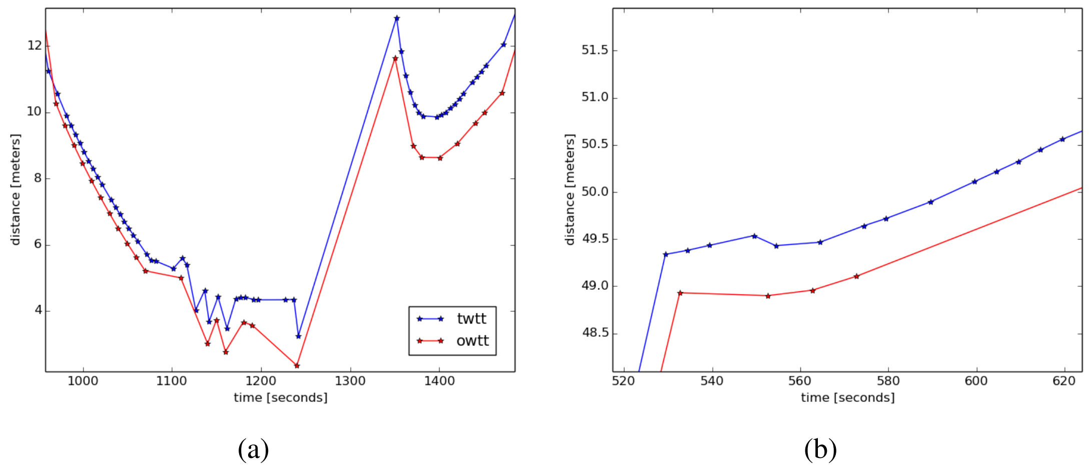


Fig. 9: Experimental comparison between distance measured from OWTT and TWTT. The experiment was conducted using a Sonobot equipped with EvoLogics modem, communicating with a static positioned modem in a small ($\sim 50 * 150$ meters) and shallow (depth < 1m) fresh water loch (Scottish lake) at Heriot-Watt University.

Almedia *et al.* have investigated this and further improved equation (4) and (6) by measuring and including the delays caused by electronics filter (t_f), processing times (t_p) and other design specific delays (t_d) [64]. The filter compensated equations for TWTT and OWTT can be seen in equation (10) and (11), which are based on the actual time measured by the system (t_{ACS}) from equation (9). In their study they achieve a result of a RMS distance error below one metre to static buoys.

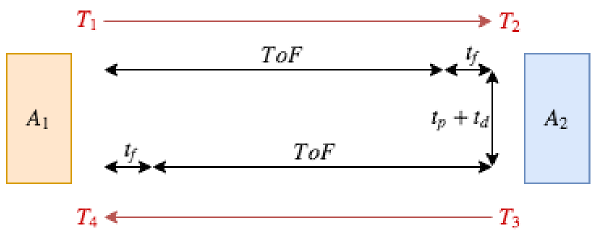


Fig. 10: Two-Way-Travel-Time with added filter, processing and other design delays. $T_{1,2,3,4}$ are the same as in Figure 8.

$$ToF = (T_4 - T_1) - (T_3 - T_2) \quad (8)$$

$$t_{ACS} = \frac{ToF}{2} + t_f + (t_p + t_d) + \frac{ToF}{2} + t_f \quad (9)$$

$$d_{TWTT_f} = \frac{t_{ACS} - 2t_f - (t_p + t_d)}{2} * v_s \quad (10)$$

$$d_{OWTT_f} = (t_{ACS} - t_f) * v_s \quad (11)$$

2.4 Vehicle Motion Model

To improve the estimations of the localisation of a vehicle over time a motion model representing the kinematics of the system can be used.

For the localisation problem of this thesis, rigid body underwater robots are considered such as a torpedo-shaped AUV. These platforms are usually equipped with a pressure sensor which can give quite accurate readings on the vehicles' depth (error within a few centimetres). Therefore, localisation can be considered to be done in 2D (Northings-Eastings, latitude-longitude or x-y) instead of 3D as the uncertainty in depth can be seen as bounded. These vehicles are most likely equipped with sensors to measure the orientation (IMU, magnetometer, compass etc.) as well as DVL to estimate velocity. As we consider the localisation problem in 2D we get the following data from the sensors: velocity(v) and yaw (θ). We can see the representation of these motion estimates in Fig. 11. Using these vehicle estimates, a first-order motion model of the vehicle can be used to describe the kinematics of the system. This can be seen in equation (12). This can be used to update the estimated position (q) from DR ($\dot{q}$) using the the delta time (Δ_t) in equation (13). A noise value is added as (N) to describe the noise from the sensors. For the work considered in this chapter we only require a simple kinematic expression of the motion model, for more in-depth descriptions there are multiple robotics books which covers this [65].

$$\dot{q} = \begin{bmatrix} v * cos(\theta) \\ v * sin(\theta) \end{bmatrix} + N \tag{12}$$

$$q = q + \Delta_t * \dot{q} \tag{13}$$

The velocity and direction can be seen as in Fig. 11. The direction and velocity of the movement does not necessarily have to be along the vehicle, as external forces (currents etc) can affect the system. v and v_θ should be seen as the velocity as well as the direction of travel and not the heading of the vehicle.

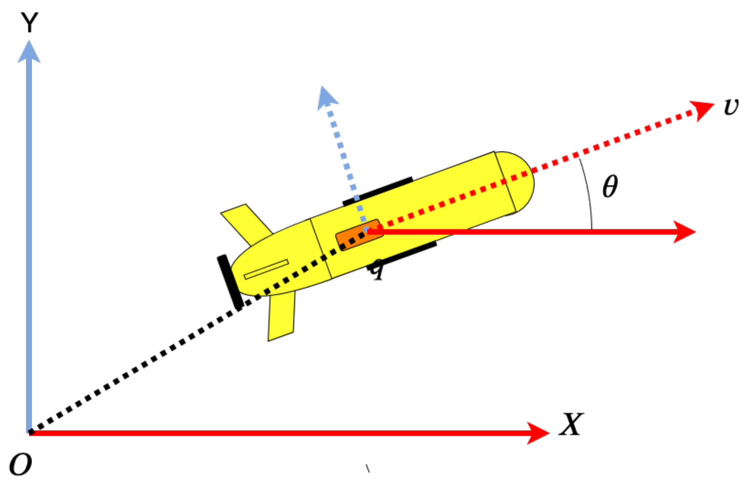
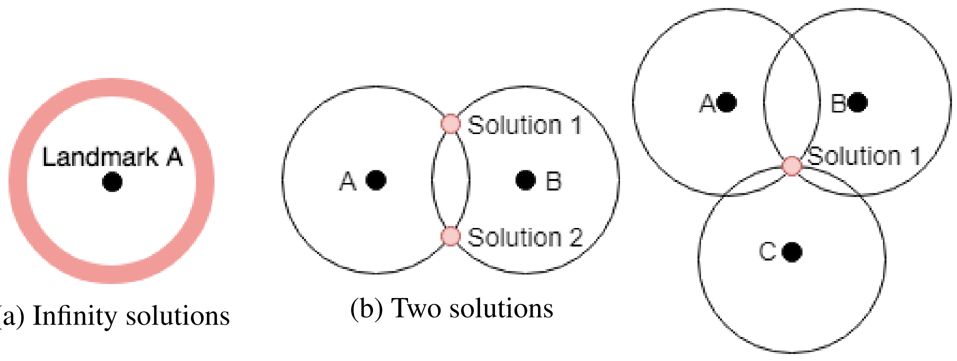


Fig. 11: The 2D kinematics from equation (12) are used to estimate a vehicle's movement from sensor data.

2.5 Localisation using Nonlinear Least Squares

If the received distance measurements, without noise are visualised as spheres in $\mathbb{R}^3$ or circles in $\mathbb{R}^2$, as seen in Fig. 12, the position of the receiver should most likely be in the intersection of the perimeter of these circles. This is the argument behind the first method for range-only localisation presented in this chapter. The method is based on maximum likelihood estimators, by solving the Nonlinear Least Squares problem. The solution of the problem is to find an estimate state where the distance to the observed landmarks are as close to the observed measurements (distances from acoustic messages) as possible, as this would correspond to the intersection of said geometrical shapes. This can be formulated as equation (14). The Nonlinear Least Squares (NLS) problem can be described as trying to fit m observation to a non-linear model of n parameters. To find a unique solution m needs to be larger than n. For the range-only localisation problem, observations are the distances measured to the landmarks and the model is the estimation of the position (North-East/X-Y in $\mathbb{R}^2$). The reasoning being that we need m to be larger than n to find a unique solution, which can be seen in Fig. 12c. Another criteria to find a unique solution is that $m > n$ observations need to be linearly independent.



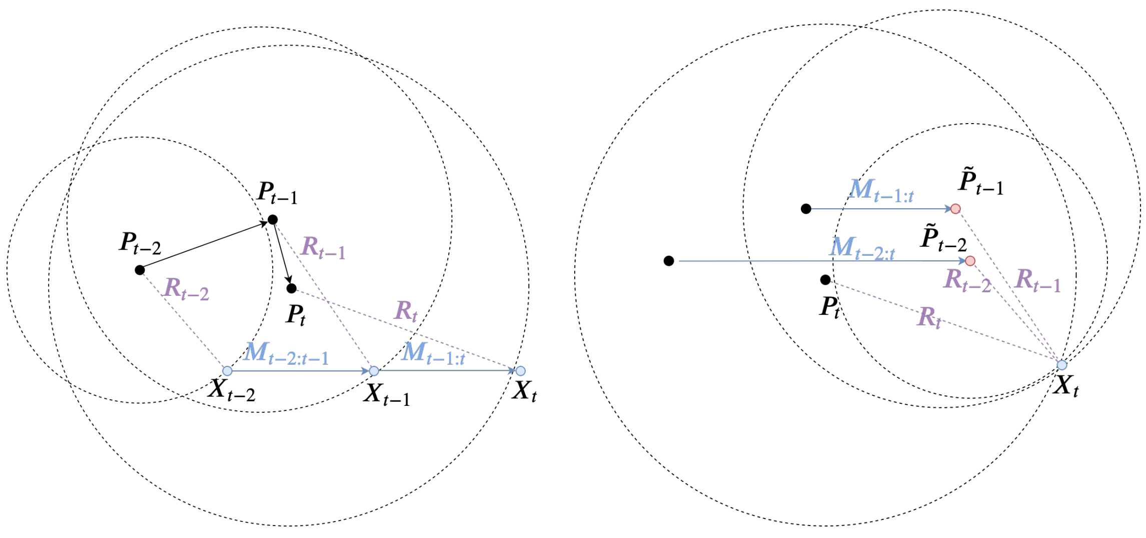
(a) Infinity solutions (b) Two solutions (c) One solution

Fig. 12: NLS for range-only localisation needs more measurements then parameters to find a unique solution. In the case of range-only localisation for finding a position in $\mathbb{R}^2$, 3 linearly independent measurements are required.

As mentioned, an AUV can be considered to be equipped with a pressure sensor. As such it is already localised in one dimension, which reduces the problem from 3D to 2D. To solve this for a unique solution, the vehicle needs to store measurements until a third one arrives. However, as measurements are received at different times, this has to be taken into account, as NLS is based on solving the problem as if all observations were received at the same instant in time. This can be performed by updating the landmarks' position along with the DR of the vehicle since the observation was made, as from the kinematic model (13). An example of how this can be performed can be seen in Fig. 13.

From a scenario such as Fig. 13, where the method uses the last 3 measurements, we get one landmark (P) and 2 virtual landmarks ($\tilde{P}$), along with their measured distances, resulting in a configuration such as Fig. 14 which can be used to find the estimated position (X^*) based on the range observation to the landmarks.

From Fig. 14, let $X \in \mathbb{R}^n$ be the current estimated position of the vehicle, where n is the dimension. As depth is considered known, the problem to solve is reduced from $\mathbb{R}^3 \Rightarrow \mathbb{R}^2$. P is the set of landmarks denoted as $P_i \in \mathbb{R}^n$ where $i \in \{1, .., t\}$. Z is the corresponding distance measurements ($Z = [Z_1, .., Z_t]^T$) derived from the Time of Flight (ToF) and R_i is the estimated distance between receiver's estimated position and the landmark P_i as $||X - P_i||$, which gives us the vector $R = r(X) = \left[||X - P_1|| \quad . \quad . \quad ||X - P_m||\right]^T$. The squared difference between estimates and measurements is then formulated as equation (14), where $E(X)$ is the cost function of the squared sum of the difference between the estimated distance at position X's distance to the landmarks and the measured distance from OWTT or TWTT.



(a) The landmarks' position (P) and the receiving platform's position (X) over time.

(b) The landmarks' position are moved according to the motion (M) of the platform since the time of reception.

Fig. 13: By compensating the movement of the landmarks (P, assumed to be localised of surface), the receiving platform can solve the NLS problem as if all signals were received simultaneously using virtual landmarks ($\tilde{P}_i$ for $i \in \{t-n, .., t\}$, where n is the number of landmarks used to solve the NLS problem.

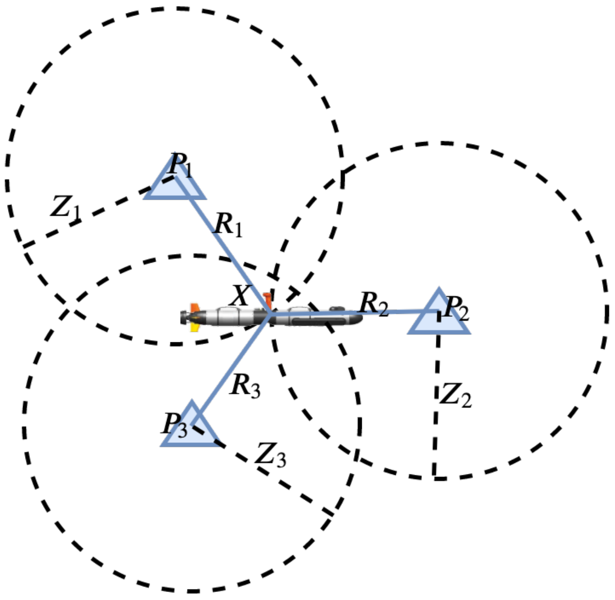


Fig. 14: An AUV can localise itself using ranges to multiple landmarks. $Z_{\{1,2,3\}}$ is the measured distance. $R_{\{1,2,3\}}$ is the estimated distance from the estimated position X to the landmarks $P_{\{1,2,3\}}$

$$E(X) = \frac{1}{2}\sum_{i=1}^{m}(||X - P_i|| - Z_i)^2 = \frac{1}{2}||r(X) - Z||^2 \tag{14}$$

Hence, to find a position X which minimises equation (14) is to find the most likely position of the receiving vehicle. We state this as:

$$X^* = \underset{X \in \mathbb{R}^n}{\arg\min}\, E(X, P, Z) \tag{15}$$

One method could be to generate a cost map of the potential X, such as in Fig. 15. In such a map, selecting the position with the lowest cost is straightforward, which is the position that minimises equation (15). However, calculating such a map involves calculating the cost in every position in the considered region, which is computationally expensive and the precision of the solution is dependant on the resolution of the map. Less computationally expensive approaches to solve the NLS problem can be performed by iterative minimisation or closed form solutions. However, in [55], Alcocer states that closed form solutions are not finding the optimal solution. Various iterative minimisation methods have been applied successfully in multiple cases including [55, 66, 67]. These approaches are used to find a minimum based on Gradient Descent (GD). There exists multiple approaches on how to use GD to solve such a problem, which will be shown in the next section.

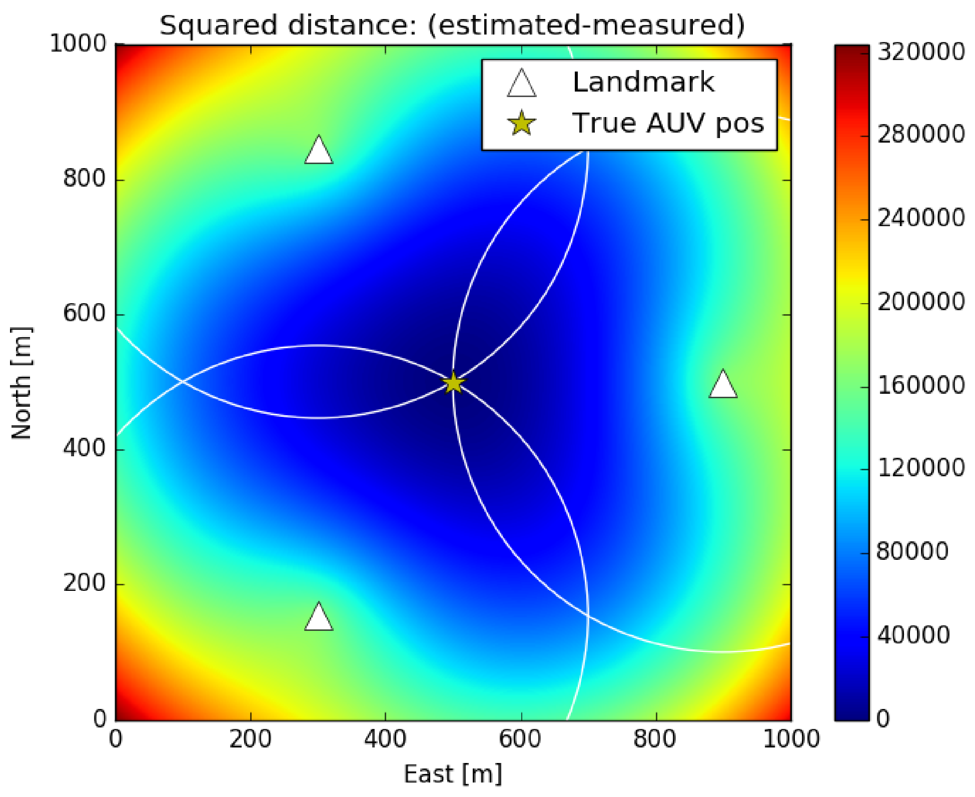


Fig. 15: Map of cost from equation (14) in an example without any noise in the range measurements (represented by the white circles).

2.5.1 Solving the NLS Problem Using Iterative Methods

Consider a scenario, such as in Fig. 16a, where a set of m ranging measurements along with the m corresponding landmarks, with known positions, are used for localisation. In the figure, a cost map has been created using (14). If the gradient at any point can be calculated, moving towards the steepest decent should result in obtaining a lower cost. By repeatedly doing this until a minimum has been found a possible solution can be achieved. To do this we need the equation of the gradient in a given point $\hat{x}$.

$$\Delta\hat{x} = \frac{\partial E}{\partial X} = \frac{\partial}{\partial X}(\frac{1}{2}||r(X) - Z||^2) \tag{16}$$

Equation (16) gives equation (17).

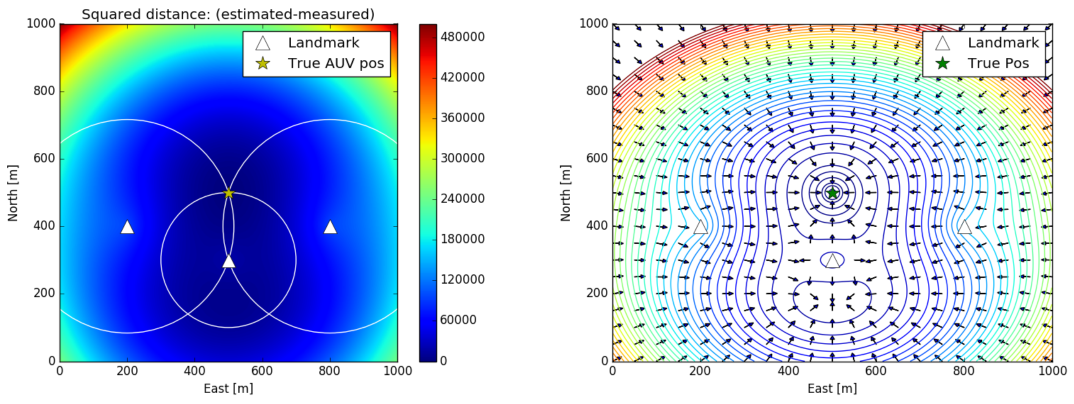
$$\Delta\hat{x} = \sum_{i=1}^{m}(\hat{r}_i)(\hat{x} - r_i) * (\frac{\hat{x}_k - x_i}{||\hat{x}_k - x_i||}) \tag{17}$$

To verify this, the negative gradients of Fig. 16a are plotted in Fig. 16b. With the equation to calculate the gradient of an arbitrary point we can now apply (17) iteratively from the initial position X_0, then moving along the negative gradient with a length dependent on λ, as in (18), resulting in a new position X_1 which should have a lower cost. This is performed until a termination condition is fulfilled. The termination condition can be based on various factors e.g. j iterations or when the magnitude of $\Delta\hat{x}$ is below a threshold, which would be either the global minimum, a local minimum or a saddle point. In Fig. 17a it can be seen how different initial X_0 reaches either a local or a global minimum.

$$X_{i+1} = X_i - \lambda * \Delta\hat{x} \tag{18}$$

The step size or learning rate, λ dictates how long X_i should move along the steepest decent. A large λ could make method converge toward a solution faster. λ should be in the range $]0, 1[$, as $\lambda < 0$ would go in opposite direction and $\lambda \geq 1$ could end up overshooting the goal and oscillating. A comparison of different λ's effect is shown in section 2.5.1.2.

An issue for gradient descent is that it cannot by itself distinguish between local minimum and global, as seen in Fig. 17a. To overcome this, Monte-Carlo Gradient Descent (MC GD) methods can be applied to instead of using a single initial point X_0, multiple initial points as $X_{j_{0,..,n}}$ can be used for the GD. Solving the gradient descent for each instance and selecting the one which results in the lower cost could reduce the chance of selecting a local instead of global minimum as the solution. An example of randomly chosen X_{j_0} for $j \in \{0, .., 6\}$ can be seen in Fig. 17 where in Fig. 17b through evaluating the cost, it is easy to distinguish between a local and a global minimum where X_j for



(a) Cost map (b) Map of the negative gradients

Fig. 16: Curvature of the cost function (14) along with the direction of the gradient from (17). The initial P will effect the gradient descent as there is a local minimum.

$j \in \{1, 2, 3, 6\}$ all result in the global minimum and the correct solution.

2.5.1.1 Variations of Gradient Descent

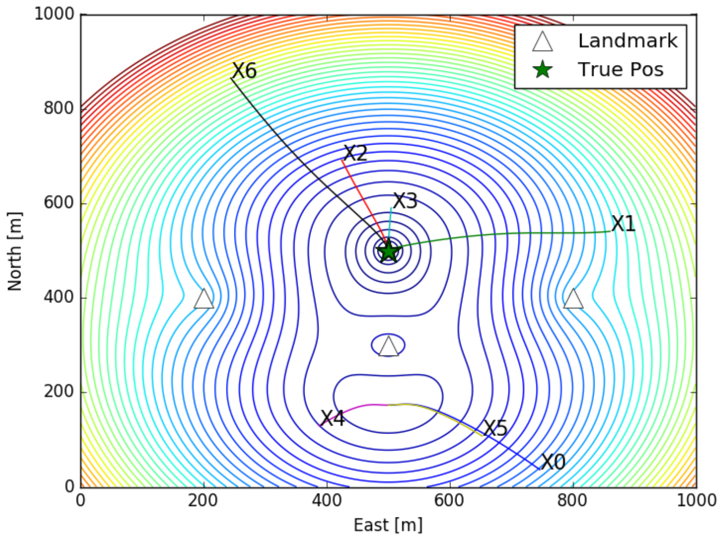
There exist many variations of GD, some which are used to escape local minimum. Examples of these are Stochastic Gradient Descent (SGD) and batch GD. In SGD, instead of using the whole set of observations in (17) a single random observation from the set of observations are used. A new random observation is selected at each iteration. Batch GD works in a similar fashion but instead of using a single observation a set of observations is used. However, for a problem with few observations it does not guarantee escaping the local minimum due to the high probability of re-selecting an observation which moves the examined point towards the local minimum. To speed up the convergence, GD with momentum can be used (Momentum Gradient Descent (MGD)). MGD make use of a weighted history of the GD to gain momentum. An example using the previous gradient can be seen in (19) where λ_{mr} is the momentum factor/rate.

$$X_{i+1} = X_i - \lambda * \Delta\hat{x}_i - \lambda_{mr} * \Delta\hat{x}_{i-1} \tag{19}$$

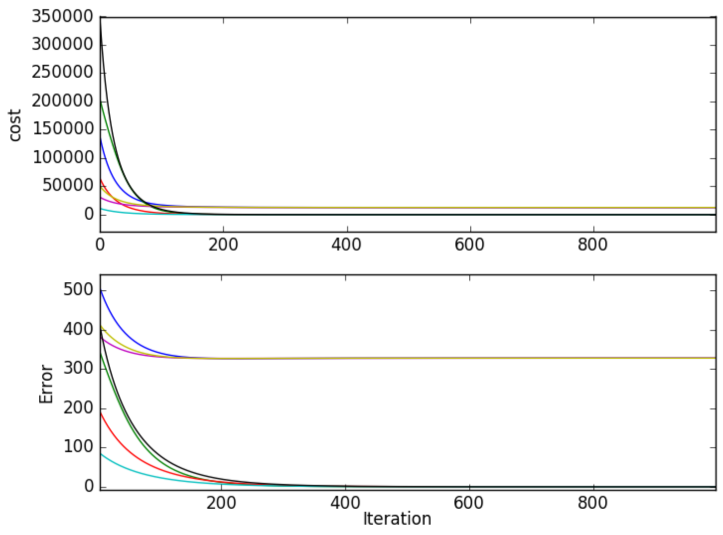
A comparison of different step size (λ) with different momentum factors (λ_{mr}) can be seen in Fig. 18b. SGD and MGD can be combined to create a stochastic approach with momentum [68]. Conjugate GD was used to solve the NLS problem in [66].

2.5.1.2 Comparison of Iterative Methods

We have discussed various approaches to solve the NLS problem using GD. This section will compare them in the terms of how fast they converge towards the solution. This will be affected both by the methods as well as the learning rate or step size along with the



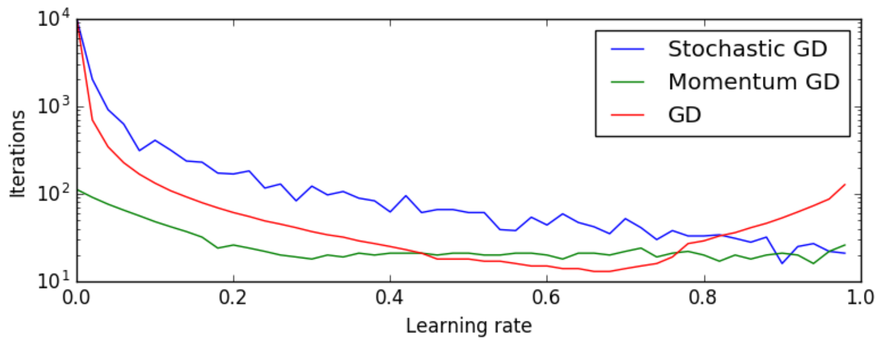
(a)



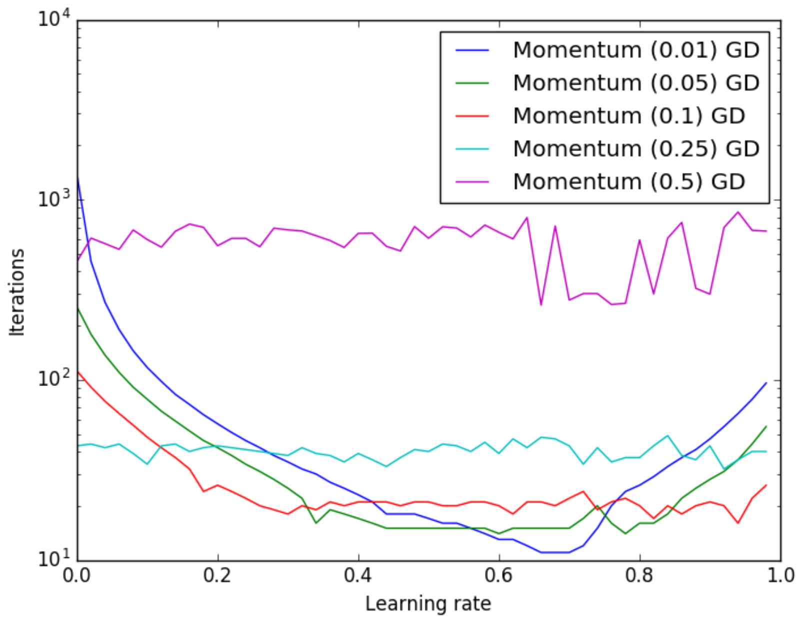
(b)

Fig. 17: Using multiple initial starting points and solving the gradient descent and choosing the solution with the lower cost is a stochastic/Monte Carlo approach to reduce the chances of selecting a local minimum. $\lambda = 0.01$

momentum rate (if used). In Fig. 18 different approaches, learning rates and momentum rates are compared in the same scenario shown in Fig. 17.



(a) MGD with a momentum rate of 0.1 compared to GD and SGD.

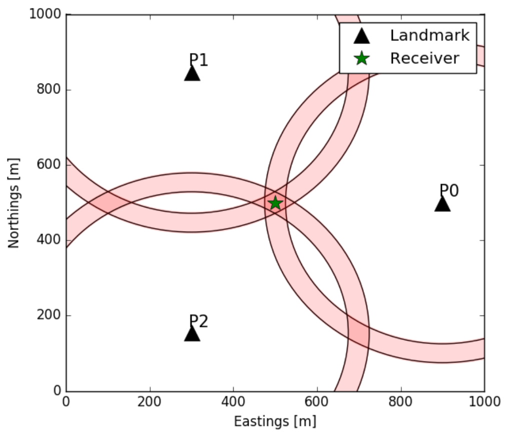


(b) MGDs with different momentum rates (value shown within parenthesis) are compared.

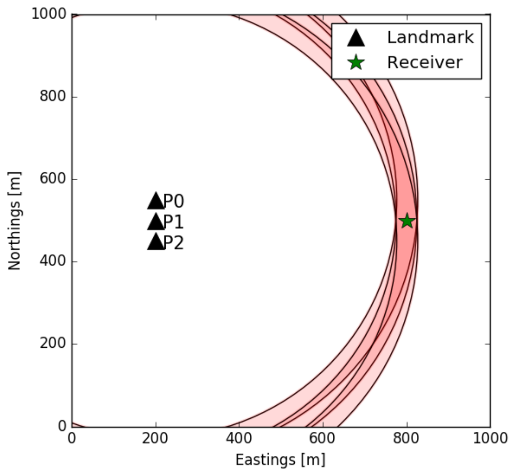
Fig. 18: The number of iterations until a solution (either when the magnitude of the gradient is $\leq 10^{-5}$ or 10^4 iterations have passed) is considered to be found for different versions of GD under different learning rates. The scenario is the same as shown in Fig. 17 with an initial point of x=200, y=800.

2.5.2 Landmark geometry and precision dilution

The error in localisation from NLS is based on the landmarks' geometrical configuration and the noise in the measured distance [28]. If we assume an equal Gaussian distribution of the noise for the distance measurement for each signal we can visualise the region of possible solutions as in Fig. 19. The optimal configuration of the landmarks in respect to the receiver is hence if the landmarks are uniformly spread in angles around the receiver [28, 69, 70].



(a)



(b)

Fig. 19: The area region of intersection (possible solutions) between the distance measurements to the landmarks is dependent on the geometrical configuration between the receiver and the landmarks.

An approach to calculate how efficient a set of landmarks is can be done through Cramér-Rao Lower Bound (CRLB) or Fisher Information Matrix (FIM), which are the inverse of each other, hence maximising the determinant of the FIM is equal to minimising it for the CRLB when it comes to optimal transmitter placement in relation to the receiver [71]. The same definition is used by Martinez *et al.* to place sensors in optimal positions based on the FIM, to track a target [69]. The FIM is also used by Glotzbach *et al.* [72] to position GIBs in optimal configurations. From the paper (20) is derived to calculate the information value. In the equation $\delta_{i,j}$ is the angle towards the landmarks in respect to the target position and σ is the noise.

$$\mathcal{L}_{q0} = \frac{1}{2\sigma^2} \sum_{i,j=1}^{n} sin(\delta_i - \delta_j) \tag{20}$$

What is of interest is that the maximisation of information value is not dependant on range, but rather solely on the incoming angles. Applying (20) to every cell in the discrete grid shown in the different scenarios in Fig. 20a, 20d and 20g, the positions where the location of the target's uncertainty would be decreased the most are the higher values (darker red). What can be clearly noted in Fig. 20a is that (20) does not distinguish between a $\delta = \omega$ and $\delta = \omega \pm \pi$. This can also be seen in Fig. 21, where the green and the red solid lines are two current transmissions' angles in relationship to the target. When finding the angle of a third transmitter which maximises the FIM as in (20), it can be seen that two solutions are found.

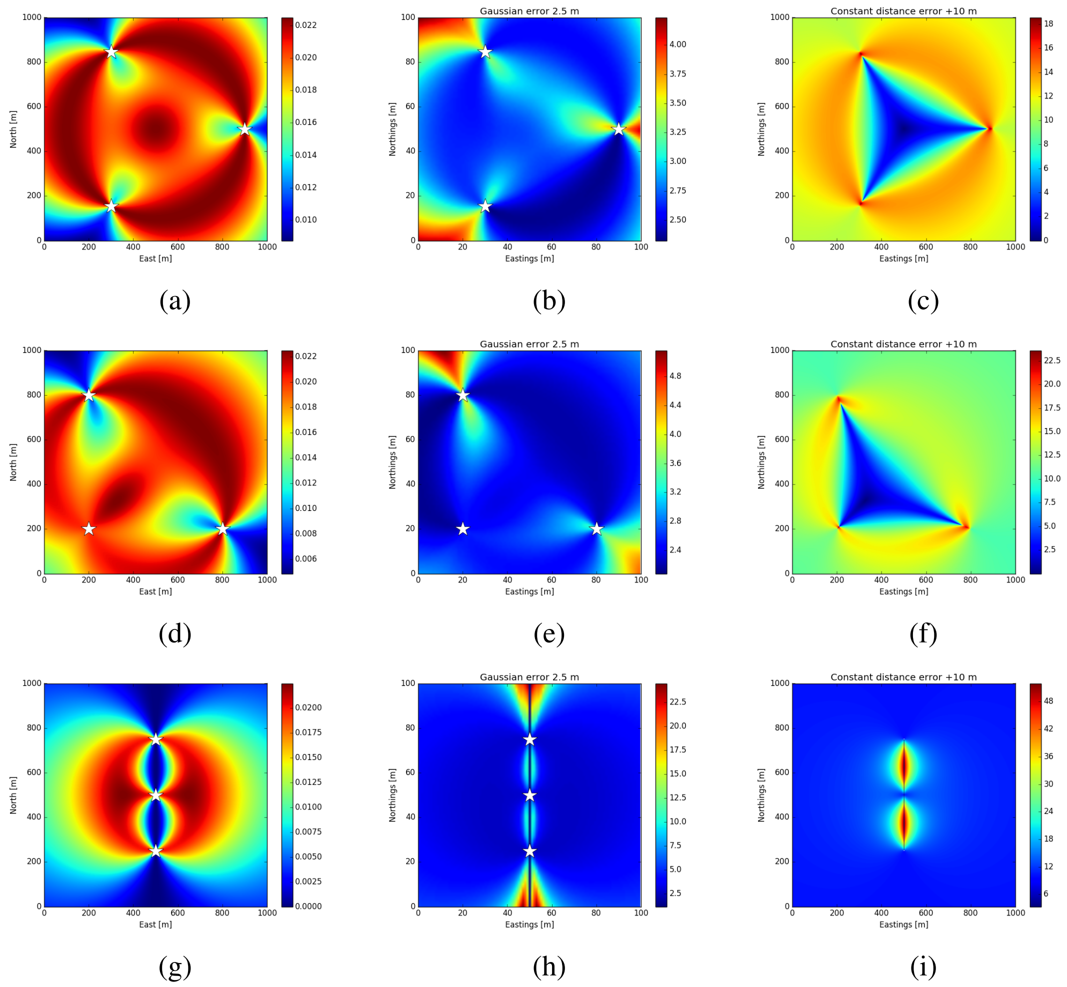


Fig. 20: Various distributions of landmarks in the different rows (white stars). The left column (a, d, g) shows the determinant of the FIM (20) in every cell. The middle column (b, e, h) shows the solution from solving the NLS problem using MGD when the distance measurement is $\mathcal{N}(True\_distance, 2.5)$. The result is the average of 30 Monte Carlo simulations for every 10:th cell. The right column (c, f, i) show the resulting error if the distance measurements are $True\_distance + 10$ metres using MGD.

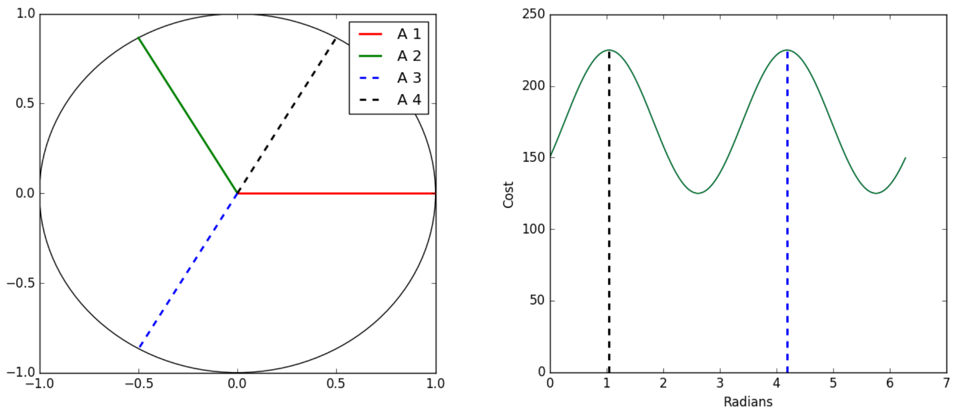


Fig. 21: The angle of current landmarks are shown as the solid lines in (a) in respect to the receiver. To find the optimal angle for placement of a third landmark, two solutions are found which maximises the determinant of the FIM as from (20). The two solutions are exactly π radians in difference.

2.6 Extended Kalman Filter

The Kalman Filter (KF) [73] is an optimal filter for linear processes with Gaussian noise which has been around since the 1960's. It is an optimal linear filter when the model is linear, the noise is Gaussian and the noise levels are known. This is clearly not the case for range-only localisation as it is nonlinear. However, there are extensions to KF to handle these non-linearities such as Extended Kalman Filter and Unscented Kalman Filter [74] among many others. While NLS requires a set of observations, these filters can use stochastic models to incorporate every measurement. The core of the filters is a two step procedure. A prediction step and an update step. The filters estimate uncertainty in the form of a covariance matrix.

The Extended Kalman Filter (EKF) has been widely used and has proven effective in the underwater environment [66, 19, 75]. Just as the KF, it is based on a prediction and an update step. The prediction step predicts the new state of the vehicle and associated uncertainty based on its current state and a motion model of the vehicle (such as described in section 2.4). The update step combines the new measurements with the predicted state to produce a new state estimate. The combination is done through the Kalman gain and takes into account the uncertainty in the predicted state and the measurement noise. When the relationships between measurements and state and/or the motion model are nonlinear, the Extended version of the Kalman Filter is used by linearising the equations around their current estimate using Jacobians.

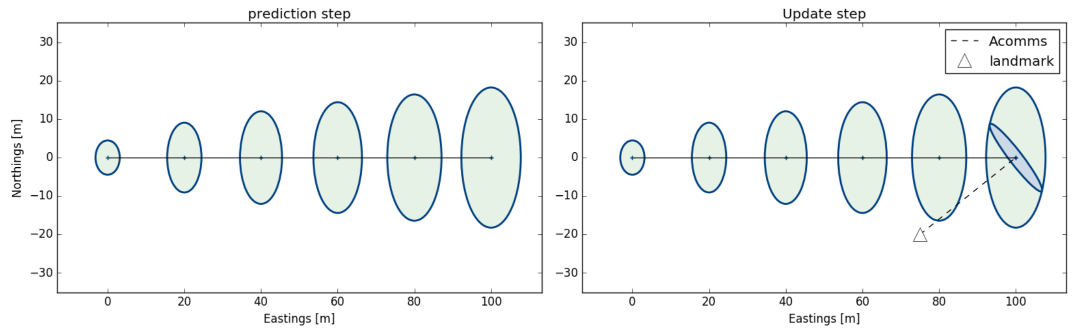


Fig. 22: The covariance matrix from the EKF are represented as ellipses. To the left, the covariance grows continuously during the prediction steps. To the right, the vehicle receives a distance measurement and updates its covariance matrix.

2.6.1 Prediction Step

The prediction step moves the state estimate according to the first-order motion model described in equation (12). This predicts the state based on the sensor reading as in equation (21).

$$\text{Predicted state estimate} \qquad \hat{\boldsymbol{x}}_{i|i-1} = f(\hat{\boldsymbol{x}}_{i-1|i-1}, \boldsymbol{u}_k) \tag{21}$$

$$\text{Predicted covariance estimate} \qquad \boldsymbol{P}_{i|i-1} = \boldsymbol{P}_{i-1|i-1} + \boldsymbol{F}_i \boldsymbol{Q} \boldsymbol{F}_i^T \tag{22}$$

The Jacobian for the state transition (25) is taken into account to update the covariance matrix (22) based on the system noise $\boldsymbol{Q}$ during the prediction step. The implementation using the Jacobian of the first-order motion model used for predication (equation (12)) can be seen in (26).

An example of how the covariance matrix grows continuously between updates during a constant motion can be seen to the left in Fig. 22. The rate of the growth is dependent on the noise matrix $\boldsymbol{Q}$.

$$\boldsymbol{F}_i = \left.\frac{\partial f}{\partial x}\right|_{\hat{\boldsymbol{x}}_{i-1|i-1}, \boldsymbol{u}_k} \tag{25}$$

$$\boldsymbol{F}_i = \begin{bmatrix} \Delta_t * sin(\Theta) & \Delta_t * v * cos(\Theta) \\ \Delta_t * cos(\Theta) & -\Delta_t * v * sin(\Theta) \end{bmatrix} \tag{26}$$

2.6.2 Update Step

The update step is used when a measurement from an external source is received. This step performs a weighted average of the current estimate and the measurement. This is performed by calculating the Kalman gain (29). In the case of range-only localisation the update step is performed when an acoustic message is received. The message needs to contain data representing the origin of the message (η) and information able to be converted to a distance (z_i).

$$\begin{aligned} &\text{Innovation or measurement residual} & \tilde{\boldsymbol{y}}_i &= \boldsymbol{z}_i - h(\hat{\boldsymbol{x}}_{i|i-1}, \eta) & (27)\\ &\text{Innovation (or residual) covariance} & \boldsymbol{S}_i &= \boldsymbol{H}_i \boldsymbol{P}_{i-1|i-1} \boldsymbol{H}_i^T + \boldsymbol{R}_i & (28)\\ &\text{Kalman gain} & \boldsymbol{K}_i &= \boldsymbol{P}_{i-1|i-1} \boldsymbol{H}_i^T \boldsymbol{S}^{-1} & (29)\\ &\text{Updated state estimate} & \hat{\boldsymbol{x}}_{i|i} &= \hat{\boldsymbol{x}}_{i|i-1} \boldsymbol{K}_i \tilde{\boldsymbol{y}}_i & (30)\\ &\text{Updated covariance estimate} & \boldsymbol{P}_{i|i} &= (\boldsymbol{I} - \boldsymbol{K}_i \boldsymbol{H}_i) \boldsymbol{P}_{i|i-1} & (31) \end{aligned}$$

In (27) and (35) $h(\hat{\boldsymbol{x}}_{i|i-1}, \eta)$ is the distance between the estimated position and the origin according to the received message.

The Jacobian for the observation $\boldsymbol{H}_i$ can be seen in (34), and the implementation specific to a 2D range-only filter in (35), where η is the location from the observed landmark (included in the acoustic message).

$$\boldsymbol{H}_i = \left.\frac{\partial h}{\partial x}\right|_{\hat{\boldsymbol{x}}_{i|i-1}} \quad (34) \qquad \boldsymbol{H}_i = \begin{bmatrix} \dfrac{(\eta_x - \hat{\boldsymbol{x}}_{i-1|i-1_x})}{(\hat{\boldsymbol{x}}_{i|i-1}, \eta)} \\ \dfrac{(\eta_y - \hat{\boldsymbol{x}}_{i-1|i-1_y})}{(\hat{\boldsymbol{x}}_{i|i-1}, \eta)} \end{bmatrix}^T \quad (35)$$

An example of how the covariance matrix is reduced when a distance measurement is received from a transponder can be seen to the right in Fig. 22.

2.6.3 Landmark geometry's effect on uncertainty

To visualise the uncertainty, the covariance matrix can be represented as an ellipse. For this thesis it will give two benefits: **1)** Better intuition on how the filter is affected by a range measurement to a landmark. **2)** The angle of the semi-major axis of uncertainty. Knowing the angle of the semi-major axis of uncertainty has great benefits when aiming to reduce the uncertainty. Just as with NLS we look at the usage of EKF in $\mathbb{R}^2$. This means that the we are interested in the 2×2 x-y part of the covariance matrix as in equation (36).

$$P_i(k) = \begin{bmatrix} \sigma_x^2 & \sigma_x \sigma_y \\ \sigma_x \sigma_y & \sigma_y^2 \end{bmatrix} \quad (36)$$

The geometrical relationship between the landmark's position (as included in the acoustic message) and the estimated position of the receiving platforms affect the update covariance matrix. The reduction of uncertainty is in the radial direction between transmission and receiver's estimated position [76]. However, as a result the uncertainty in the tangential direction is not affected. As such, the reduction is reduced by the most if the position of a transmission is along the semi-major axis of uncertainty [29, 57]. This is illustrated in Fig. 23, where it can be seen how different geometric configurations affect the uncertainty.

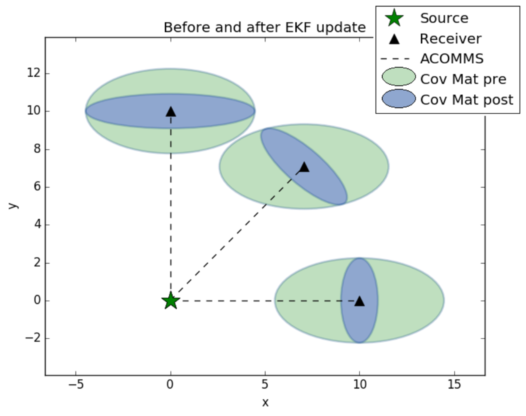


Fig. 23: The EKF's covariance matrix before (green ellipse) and after (blue ellipse) an update has been received over acoustic communication. The reduction is an effect of the geometrical configuration of the landmark and the receiver. The closer to the receiver's semi-major axis of uncertainty the message is transmitted from the more the area of uncertainty is reduced.

2.7 Particle Filter

The Particle Filter [77] is based on Monte Carlo techniques [78], which are designed to simulate the measured parameters as many instances. While EKF approximates its state around one point and Unscented Kalman Filter (UKF) usually around $2*n+1$ (for $\mathbb{R}^n$) the Particle Filter (PF) consists of an arbitrary amount of particles. Each particle represents a possible state (δ) with an assigned weight (ω). The weight represents how well that particle corresponds to a measurement. Much like the Kalman family, PF is based on a prediction and update step but with an added re-sampling step. It is a powerful filter, able to estimate various parameters. However, the amount of particles needed grows exponentially with the dimension/number of parameters. The performance and computational load is tightly linked to the amount of particles: the more particles the higher likelihood to represent the true position correctly, however more computations are needed in each step of the filter making it computationally more expensive.

2.7.1 Prediction Step

The particles are moved as a specified model with a control input, using (12) with a Gaussian noise to update each particle P_i as in (37). This noise (ω) is necessary to make the approach probabilistic instead of deterministic, as the particles would all otherwise converge to the same estimate. In (37), v is the measured velocity and Θ the heading.

$$\begin{bmatrix} P_{i_X} \\ P_{i_Y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \end{bmatrix} * \begin{bmatrix} P_{i_X^-} \\ P_{i_Y^-} \\ cos(\mathcal{N}(\Theta,\, \sigma_\Theta^2)) * \mathcal{N}(v,\, \sigma_v^2) \\ sin(\mathcal{N}(\Theta,\, \sigma_\Theta^2)) * \mathcal{N}(v,\, \sigma_v^2) \end{bmatrix} \tag{37}$$

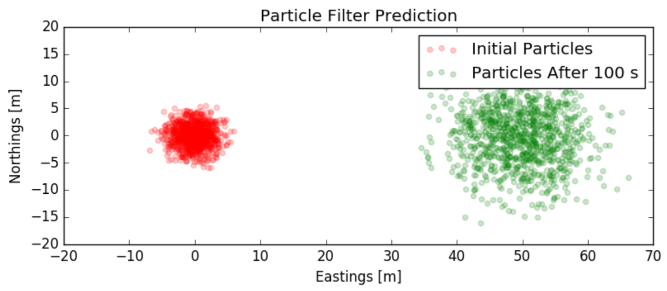


Fig. 24: The prediction step of PF moves the particles according to (37). In this scenario the distribution can be seen after 100 seconds, with a $\sigma_\Theta = 0.1$ and $\sigma_v = 0.05$ during a movement at 0.5m/s.

2.7.2 Update Step

The update step of the filter is applied when an acoustic localisation message is received. This step updates the weight of each particle to represent their likelihood to the real state according to a model. Bayes theorem is used to assign the weight of each particle as in equation (38).

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)} = \frac{likelihood \times prior}{normalisation} \tag{38}$$

For range-only localisation, the update step is to assign the weights as in (38) is based on how well the measurement matches the estimate. This is done by a Probability Density Function (PDF). The weights are then normalised so that (41) holds.

2.7.3 Re-sample Step

After the update step has taken place, the set of particles are evaluated to see if they need to be re-sampled. This is to re-sample particles with low possibility to represent the system.

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} \omega_i^2} \tag{39}$$

If $\hat{N}_{eff}$ falls under a threshold (conventionally $< N/2$), a subset of the particles are re-sampled. The re-sampled particles are randomly sampled among the remaining particles with a bias towards the ones with low probability (weights). The particles that have been selected for re-sampling are then re-sampled as a copy of one which was not selected for re-sampling (with a bias towards the ones with high probability (weight)).

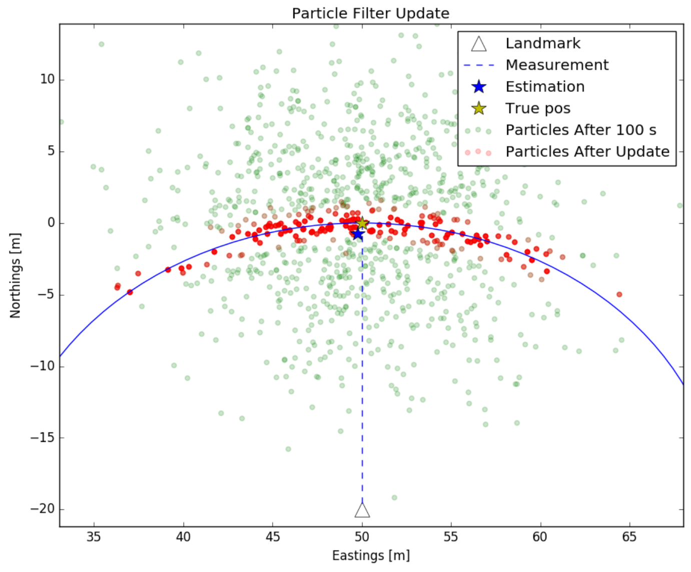


Fig. 25: The update step of the PF where the noise estimate of the signal is 0.5 metres. Following the update, the particles with a low weight will be deleted and re-sampled as a copy of a particle with a high weight. The weight is dependent on how well the particles fit the measurement, which is greater the closer to the circumference of the circle with the measured distance with its centre at the landmark.

2.7.4 State estimation:

The state estimation is based on a weighted average from all the particles as in (40). For range-only localisation this might cause bad initial estimates based on the shape and spread of the particles. This can be seen in Fig. 25 where after the first update (and re-sample) step t gives an estimate which does not necessarily represent the true position well. However, after a second one from another geometrical configuration, the state estimate will represent the true position better.

$$X = \frac{1}{N}\sum_{i=1}^{N} \omega_i \delta_i \tag{40}$$

$$\sum_{i=1}^{N} \omega^i = 1 \tag{41}$$

2.7.5 Landmark geometry's effect on uncertainty

The update and re-sampling step for the described PF results in keeping the particles that lay close to the perimeter of the circle with the measured radius (from ToF) with the centre at the landmark. As such the shape of the resulting particles will be dependant on the distance to the landmark (and its position) and the spread of the particles. The further away the more of a straight line the particles will result in, and the closer it is, the more curvature the shape of the particles will have. A curvature shaped spread results in a poorer estimate, as it will have a greater variance in two dimension compared to a line, which might have a high variance in the perpendicular direction to the vector between the estimates and the landmark but low variance in the other dimension. The lower the variance, the more the particles should represent the true state. As such, to lower the variance, and hence in theory improve the state estimate, the observed landmarks should preferably vary geometrically in a way which reduces the variance the greatest. Assuming that the particle spread is mainly tangential to the angle towards the landmark, it is intuitive that the next transmission should have a 90° difference to reduce the spread of the particels to the greatest extent. Fig. 26 shows the effects from different geometrically distributed landmarks on the same particles. The previous observed landmark was in the negative north direction (as in Fig. 25). Therefore in theory a landmark observed from either positive or negative in the straight east should reduce the variance the greatest. From Fig. 26 it can be seen that this statement holds.

While the state estimation after an update for PF looks different to the covariance matrix in the filter Kalman filter, the *angle* (of the cluster of pixels) and the variance of the particles are similar to the ellipse of the covariance matrix. The geometry between landmark and the receiving AUV effects the density of the re-sampled particles after an update in a similar fashion as EKF. As the particles become more dense, the observation is to the line upon which one would consider the major axis of uncertainty. This can be seen in Fig. 26. In the lower right subfigure in this figure, it can be seen that sequential updates from the same angle reduces the density, and hence the uncertainty, by the least, while a 90° angle difference has the most benefits.

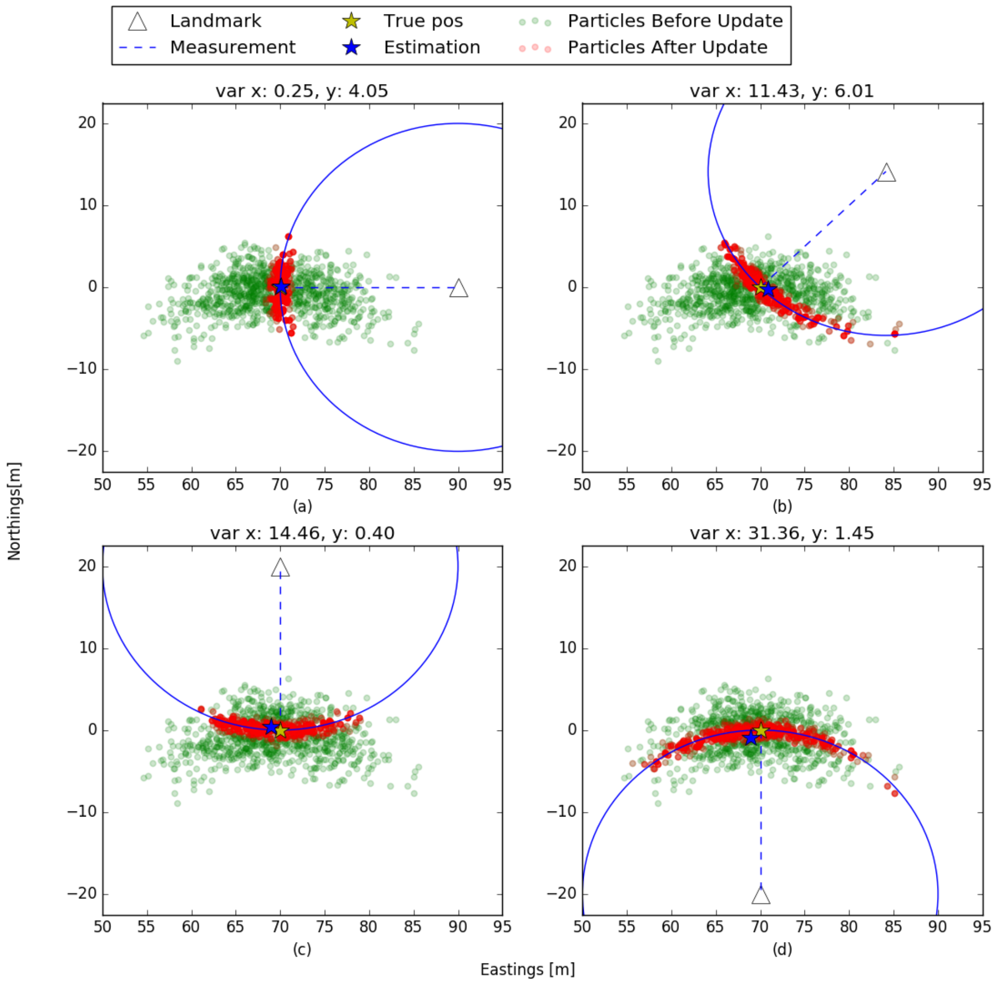


Fig. 26: The density of the particles after an update and re-sampling step is dependant on the particles before the update and the geometry with the landmark. The particles in this scenario are from Fig. 25, after they have travelled 40 more seconds. Var is the variance in x and y compared to the mean of the estimation of the state from the particles.

2.8 Localisation Filters and Methods Comparison

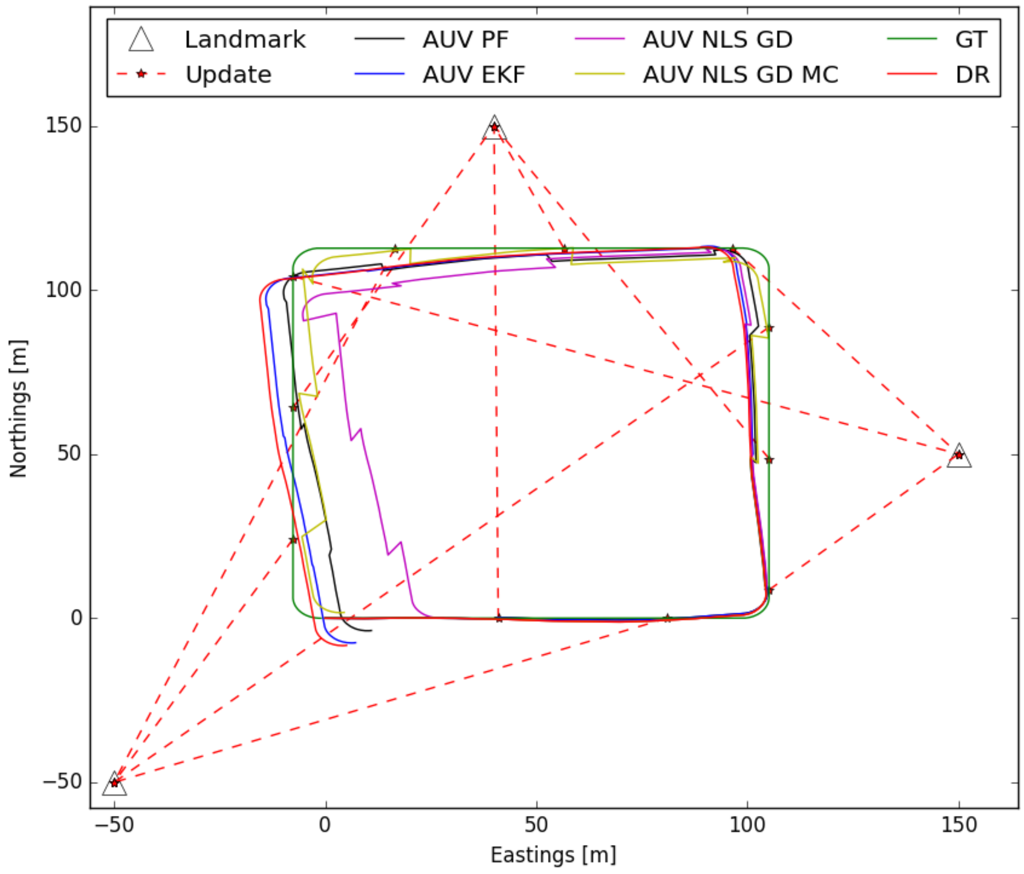
This chapter has presented three commonly used methods and filters used for localisation. They have all seen some usage in the underwater domain for range-only localisation and have their own benefits and drawbacks when it comes to precision and computational load. In this section the different methods are compared using both simulated data and data collected by AUV Sirius. We look at this from different configurations of static transponders with different types of error in the distance measurement and see how this is affected by how the number of transponders create different geometrical relationships over time.

Previous comparisons of different methods for range-only localisation methods for the maritime environment have been performed in:

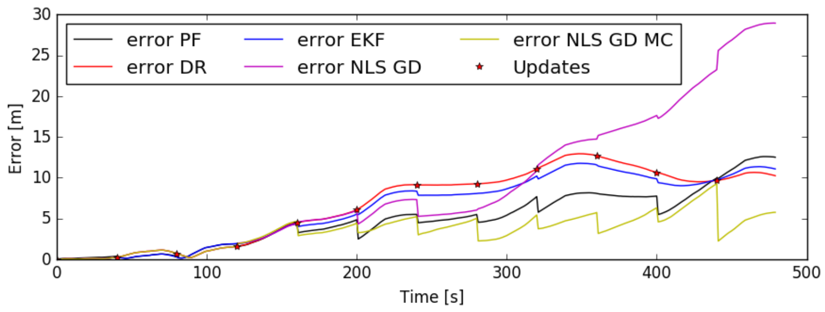
- Fallon *et al.* compares NLS (online and post processed), EKF and PF. Their results and conclusion regarding this is that NLS performs the best, while EKF the worst. The same conclusion was obtained in [66] where a conjugate GD method was used for the NLS problem.
- In [79], Murphy and Hereman compare NLS to Linear Least Squares and solve the trilateration through linearised equations for localisation in mines. The result shows that NLS is the better method, which is also stated by Alcocer [55] when compared to closed form solutions.
- [80] compared UKF and EKF for a short in-water test, but the results are too similar to draw any definitive conclusions.

2.8.1 Simulated Scenarios - No Constant Error

In the following simulated scenarios, an AUV is following the edges of a square. It runs with a speed of 1.0 metres per second while moving forward, and 0.5 while turning. The distance measurement is $\mathcal{N}(True\_distance, 0.5)$ metres. It shows from Fig. 27 and 28 that the results agree with previous comparisons: NLS performs the best, followed by PF and EKF which has the highest average error. The comparisons have been evaluated using 100 Monte Carlo simulations.



(a)

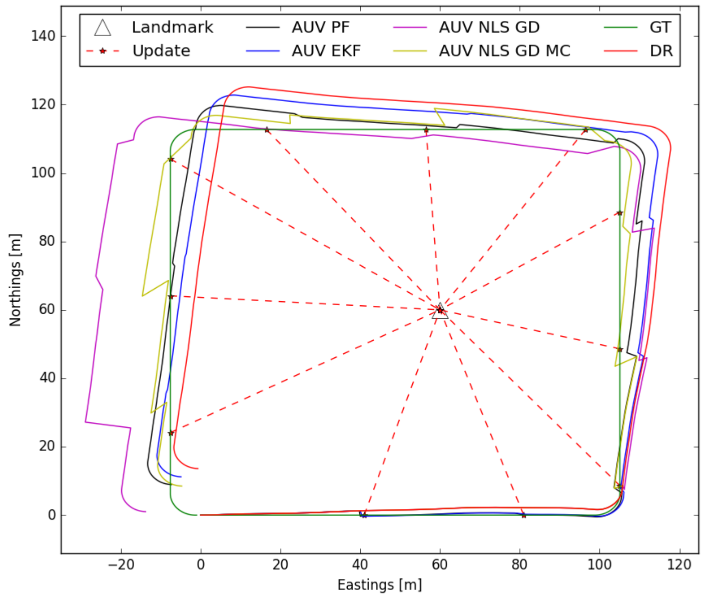


(b)

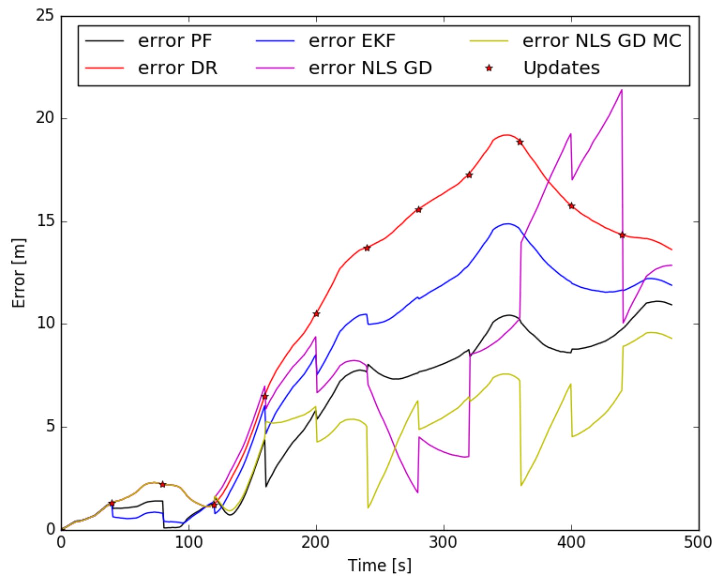
Method	PF	EKF	NLS GD	NLS MC GD	DR
Error [m]	9.2	10.5	12.7	8.2	11.3

(c) The average error for 100 Monte Carlo simulations

Fig. 27: A comparison between localisation methods using 3 static transponders transmitting in a round robin fashion periodically. An AUV is moving in a square. The AUV is measuring the following with a noise in a Gaussian distribution around zero with the following standard deviation: Velocity with 0.5%, Heading with 1.0% and distance with 0.5 metres. Both forms of GD use from 3 up to 6 of the most recent measurements, MC GD uses 10 instances of GD and chooses the solution with the minimum cost. The PF uses 1000 particles. The jumps in localisation is due to the estimate of the vehicle is assigned as the estimate from the filter. No path smoothing was performed.



(a)



(b)

Method	PF	EKF	NLS GD	NLS MC GD	DR
Error [m]	7.8	9.2	9.3	4.6	11.2

(c) The average error for 100 Monte Carlo simulations

Fig. 28: A comparison between localisation methods using a single static transponder. An AUV is moving in a square. The AUV is measuring the following with a noise in a Gaussian distribution with the following standard deviation: Velocity with 0.5%, Heading with 1.0% and distance with 0.5 metres. Both forms of GD uses from 3 up to 6 of the most recent measurements, MC GD use 10 instances of GD and chooses the solution with the minimum cost. The PF uses 1000 particles. The jumps in localisation is due to the estimate of the vehicle is assigned as the estimate from the filter. No path smoothing was performed.

2.8.2 Simulated Scenario - Constant Error

The acoustic channel is hard to model, it changes with depth and is prone to multi-path from the surface, bottom and other obstacles. As such, the truth is that the error in distance measurement might not always have a Gaussian distribution around the actual distance but instead have constant errors due to wrong channel model to estimate the speed of sound in water [81]. With this in mind it is important to see how the filters perform when they have a distance measurement infused with a constant bias (ε) added to the distance measurement $\mathcal{N}(True\_distance, 0.5)$.

2.8.2.1 3 transponders

The following table describes the error in same scenario as in Fig. 27 while the distance measurement has an additional constant error of +10 metres.

Method	ε	PF	EKF	NLS GD	NLS MC GD	DR
Error [m]	2.5	7.5	5.3	9.2	5.2	9.4
Error [m]	10.0	10.4	11.5	18.2	8.1	11.7

Tab. 3: Localisation comparison when a constant error of 10 metres is added to each distance measurement. In this case the same configuration between the 3 transponders as in Fig. 27 is used.

2.8.2.2 Single transponder

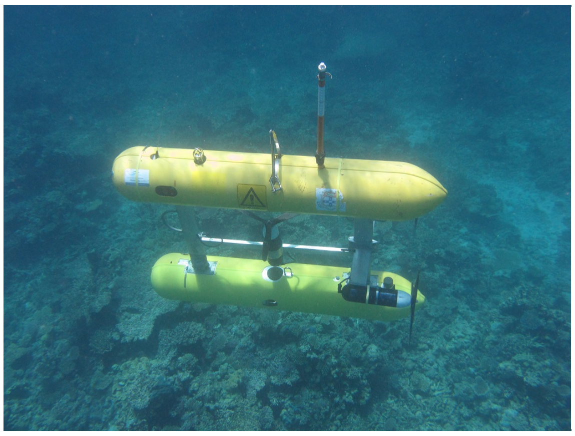
The following table describes the error in same scenario as in Fig. 28.

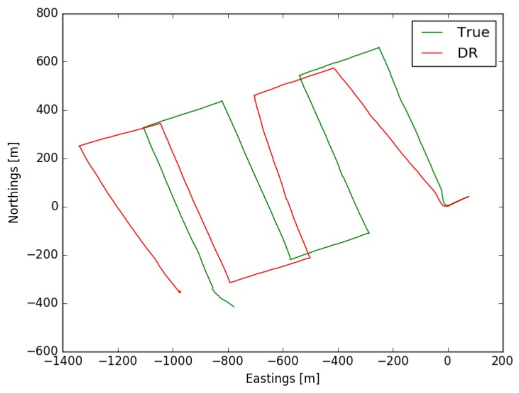
Method	ε	PF	EKF	NLS GD	NLS MC GD	DR
Error [m]	-2.5	7.0	6.7	12.0	6.4	10.7
Error [m]	2.5	9.0	7.9	13.6	7.8	11.1
Error [m]	5.0	11.6	9.1	22.3	9.1	12.9

Tab. 4: Single transponder with constant added error ε Fig. 28

2.8.3 Navigational Dataset from AUV Sirius

The following comparison is based on simulated acoustic transponders and a navigational dataset from AUV Sirius. AUV Sirius[82] is a vehicle owned and operated by Australian Centre for Field Robotics, University of Sydney. They have provided a navigational data set which has been used in various scenarios in this thesis. The dataset is divided into two used parts, the first is the raw navigational data used for DR. The second uses the DR in combination with USBL and vision based SLAM[16]. The length of the mission is roughly 3 hours. The vehicle and the path (DR and the second part which is considered to be the true path throughout the thesis) can be seen in Fig. 29. The average error from DR for the AUV is 195.4 metres.


(a) AUV Sirius. Image-credit ACFR.

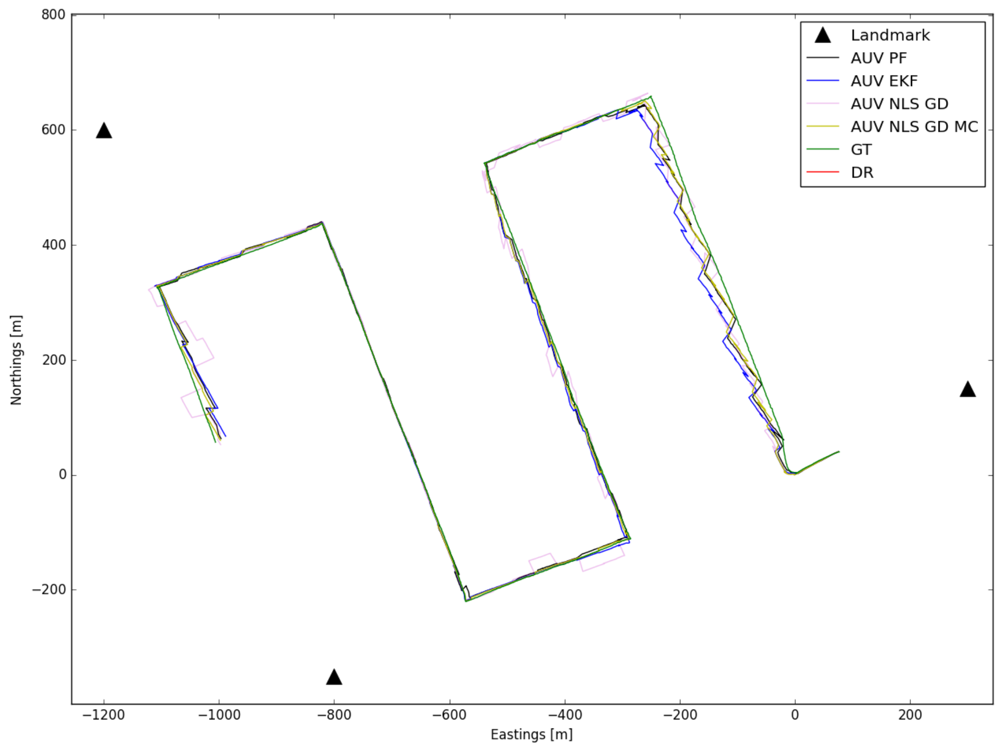


(b) Path from dataset.

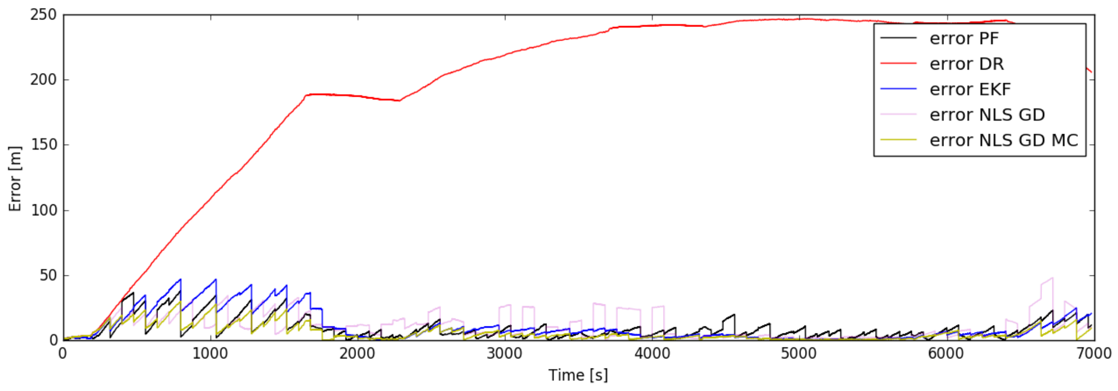
Fig. 29: A dataset collected by AUV Sirius outside of Tasmania, Australia has been used for navigational data to test and compare different localisation methods.

2.8.3.1 Multiple transponders - no constant error

The following scenario uses 3 static transponders, as seen in Fig. 30a, which transmits in a round robin fashion. There is no added constant noise. There are different simulations with different $\sigma_{distance}$ when the distance measurement is $\mathcal{N}(True\_distance, \sigma_{distance})$.



(a) Resulting localisation estimates with no noise in the distance measurement.



(b) Positional error from (a)

$\sigma_{distance}$	PF	EKF	NLS GD	NLS MC GD
0.0	8.5	10.1	10.7	5.5
1.0	8.5	10.1	24.2	5.4
2.5	10.2	14.4	37.8	6.8
5.0	11.5	22.2	43.6	9.2

(c) The average error for the different methods using different levels of noise in the distance measurement. The DR error is 195.4 metres. The jumps in localisation is due to the estimate of the vehicle is assigned as the estimate from the filter. No path smoothing was performed.

Fig. 30: The different localisation methods compared using different distributions of the noise around the true distance.

2.9 Communication and Navigation Aid

The concept of a Communication and Navigation Aid (CNA) is relatively new and was first coined by Vaganay *et al.* in 2004 [20]. The purpose was to try the concept of Moving Long Baseline (MLBL), a method in which one or multiple vehicles act as navigational aids to other vehicles by transmitting data used for localisation. Having a surface vehicle to act as an CNA have multiple benefits. Firstly, an ASV has constant access to GPS making it a reliable beacon to transmit localisation messages as its position is accurate within a few meters at all time. Second, having a vehicle with access to multi-modal communication (acoustics, satellite, wi-fi etc.) brings the possibility to achieve a close to real-time communication link with submerged AUVs from a Command and Control Centre (CCC) (which could be over the horizon), as the CNA can relay the acoustic communication to other mediums as seen in Fig. 31. The benefits are well discussed by researchers at Woods Hole Oceanographic Institution (WHOI), where they present how using ASVs/CNAs in collaboration can remove the need for ships during long term missions [83]. The term CNA has in literature also been used for AUVs with high confidence in their localisation, but using an AUV as a CNA removes the possibility for multi-modal communication while submerged. There are multiple benefit of using CNAs compared to for example LBL. First, having moving acoustic transponders which can be used for localisation can overcome the shortcomings of LBL only working in a limited region. This is as the CNA can *follow* the surveying AUVs. The second, is as stated earlier, the possibility to use multi-modal communication to relay data. The deployment and calibration of LBL is a time consuming task, while a CNA can operate autonomously. Since the vehicles are operating in the same area, the surface vehicles also have the possibility to launch and recover the AUVs [84, 85, 86]. Which can reduce human interaction even further.

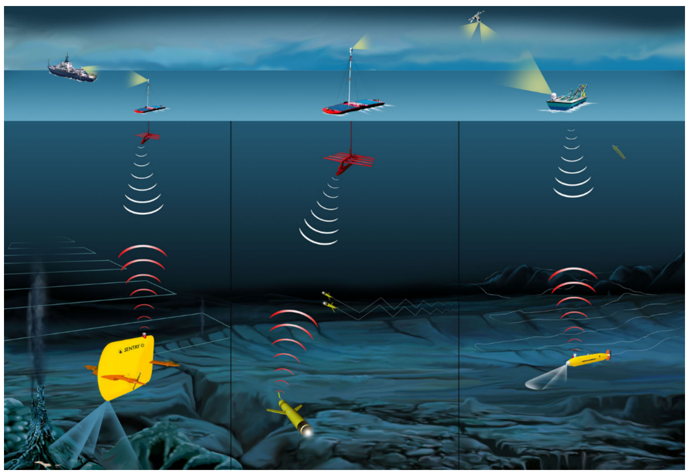


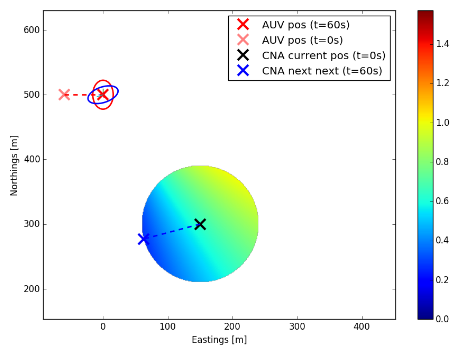
Fig. 31: Having a CNA on the surface can relay acoustic communication to and from a AUVs over the horizon or to other platforms not in close proximity as well as aid with localisation data. Image-credit [83].

A natural progression for cooperative maritime robotics is to reduce the amount of resources needed to perform the same objective. As such, for cooperative ASV-AUV localisation, only using a single CNA/ASV to perform single-beacon localisation would be beneficial. Reducing the number of vehicles reduces the cost and time to deploy the system, while as a drawback also reduces the variety of positions to transmit localisation messages from, and as such more sophisticated planning on the ASV is generally desired. Fallon *et al.* [22] used a set of repeating motion patterns on a CNA to encircle or zigzag above the surveying AUV while transmitting acoustic messages periodically every 10 seconds. As in [56], a comparison between the mean positional error on the AUV is performed using EKF, PF and NLS (online and post-processed). This comparison is also made in [66]; the observability of the system is discussed in both. The results of the comparison between the three mentioned localisation methods all show similar conclusions, which is that NLS is the best performing filter, especially when post-processed [87, 22, 66]. While the results shown in these papers manage to bound the positional error on the receiving platforms, no intelligent planning was performed on the CNA. Either it was a simple case of leader-follower scenario with and offset, encircling or zigzag pattern. In 2010, Chitre and Gao [88, 76] presented a motion planning algorithm which has the objective to minimise localisation error on AUVs. The CNA used transmits periodically with the aim to transmit from the semi-major axis of uncertainty on the AUVs, which uses EKF to localise itself[1]. The path is found using Dynamic Programming (DP), with the paths of the AUVs known *a priori*. An option to compute the plan online as a response to received information from the AUVs is available. Teck uses cross-entropy to learn a planning policy for a CNA to bound the error on AUVs. This policy is however learned considering static speeds on the AUVs [89]. Quenzer and Morgansen control the helming of a CNA based on the expected observability on AUVs [90]. Bahr *et al.* also aim to transmit from the semi-major axis of uncertainty of an AUV. By using the maximum speed of the CNA and the time until next transmission they create a cost map (in the reachable region of speed$*$time) around the CNA and then selects the position which is closest to the semi-major axis of uncertainty for the AUV [29], which can be seen in Fig. 32.

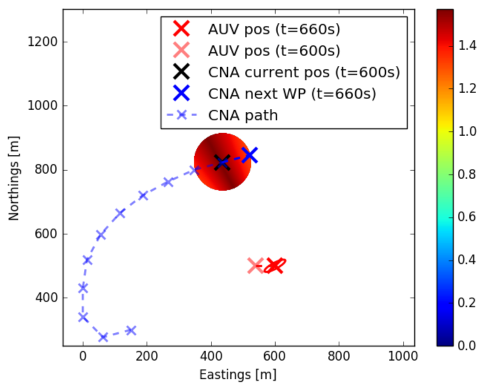
Glotzbach *et al.* [72] use the determinant of the FIM, see (20), as a cost function to find the optimal transponder placement. Similar, Munafó *et al.* use the approach of sampling and evaluating the FIM, where the sampling approach is on the perimeter of an circle which is based on both vehicle speed and the sea current [71].

Yan and Chen *et al.* [91, 70] look at optimal configuration of multiple ASVs (3+) in position based on both polar angle and distance, to optimise transmission position based on distance to the estimated position of an AUV. Their cost function to find distance is based on the distance measurement error and the depth of the AUV. In the latter paper, how the number of surface vehicle affect the localisation error on the AUV is compared, to no surprise, the more vehicles available to transmit, the more variance in measurements which results in a lower error. It is interesting how this contradicts statements in papers

[1]This is the geometrical relationship which reduces the uncertainty the greatest in EKF, see section 2.6



(a) Initial configuration.



(b) After 10 minutes.

Fig. 32: The path-planner presented in [29] evaluates the reachable region around the CNAs to select the best position to transmit a localisation message based on the received estimates from the AUVs. In this scenario the CNA transmits periodically every 60:th seconds with a maximum speed of 1.5 m/s. The AUV moves with 1 m/s, and its covariance matrix can be seen represented as the red ellipse. The colour bar represents the residual angle between the vehicles and the semi-major axis of the covariance matrix (red ellipse). The planner selects the position which results in the lowest residual angle as the next waypoint. As seen in (b), the CNA does not have a big range of angles towards the semi-major axis to choose from. This is an issue with planning only a single waypoint ahead in time.

such as [72] and [69] where the opposite is stated, that only the polar angle affects the FIM, and hence the distance should not affect the outcome. However, during simulations with uniformly distributed (120° apart) beacons and distance measurements with added noise and/or constant error the same conclusion could not be made as the distance did not show any particular affect on the solution. However, this might bring benefits during deep sea operations as further away (in the X-Y plane) creates a larger slant angle.

2.10 Summary

This chapter has given an introduction to acoustic communication in the subsurface domain and how it can be used for localisation. The methods for localisation are based on distance measurements between a transmitter and a receiver, which is done by measuring ToF. While other methods such as measuring angles (USBL or SBL) or TDoA exist, they often require multiple exchanges of messages or that the receiver is static. OWTT (using ToF) can instead support all receiving platforms with distances from one single measurement and can hence be desirable in many cases of multi-vehicle operations. There are multiple approaches to solve the distance-or range-only localisation problem. In this chapter the more common families of methods and filters used today are presented along with the current state of the different methods and a comparison between the presented methods. A focus for this thesis is to improve range-only localisation by positioning of mobile transponders. This is something that is discussed in many works focused on using CNAs. In accordance with this objective, the geometrical configurations which help improve the result for the different localisation algorithms are described.

The results from simulations and conclusions from other papers clearly state that NLS is the better localisation method for range-only measurements, followed by PF and then EKF. However, PF's accuracy is highly dependent on the amount of particles, and the more particles the greater the computational time, which for small embedded systems might not be optimal. However, PF is able to represent the possibility of multiple solution if the distribution is multimodal. Solving NLS through iterative methods (which are shown to be the better ones compared to closed form solutions [55]) can end up with a solution in a local minima and depending on factors including the geometrical setup of landmarks and the learning rate, might require many iterations to find a solution which is the same drawback regarding computational cost on embedded systems as for PF. As such, the filters of the Kalman family should not be ruled out as possible solutions.

3 Path Planning

"Space and time are intertwined;
we cannot look out into space, without looking back into time"

- Carl Sagan

The previous chapter discussed how a robot can localise itself. This is one part of navigation, the other part is how to plan and move the robot. The purpose for a robot to move depends on its objective. But no matter what the objective is, it is important that the trajectory is collision free to ensure safe operations.

Path planning is the art of finding a path which fulfils specified requirements. The most common being start-to-goal planning: assume start configuration q_{start}, finding a collision free path to X_{goal}. This can be defined as in equation (42), to find the set of actions ($U_{1,..,n}$) which leads the robot from the start configuration to the goal under some potential *parameters*. An example could be to find its way to the centre of a maze as in Fig. 33.

$$U_{1,..,n} = f(q_{start}, X_{goal}, parameters) \tag{42}$$

Another category of path-planning is application or objective based, where the goal is to maximise the value of a path based on specified criteria. This could include area-coverage [92, 93], leader-follower scenarios [94], avoiding collision at sea (COLREGs) [95] or as one of the major objectives for this thesis, to plan a path which can support other vehicles or platforms.

Application based path planning often builds on extensions of the same algorithm as start-to-goal. Therefore, this chapter will present some of today's most used forms of path planning algorithms where the objective is to take a vehicle from one configuration to a goal. The optimal solution is generally considered to be the shortest path, which can be analytically derived for simple problems. However, as the dimension and complexity of the problem increases so does the complexity of the analytic approach. Due to the complexity of the system and environment, analytic solutions are not seen as viable. To solve this navigation problem, path-planning has become a well-researched topic, leading to a sea of possible approaches. Some which are inspired by nature (Ant Colony Optimisation,

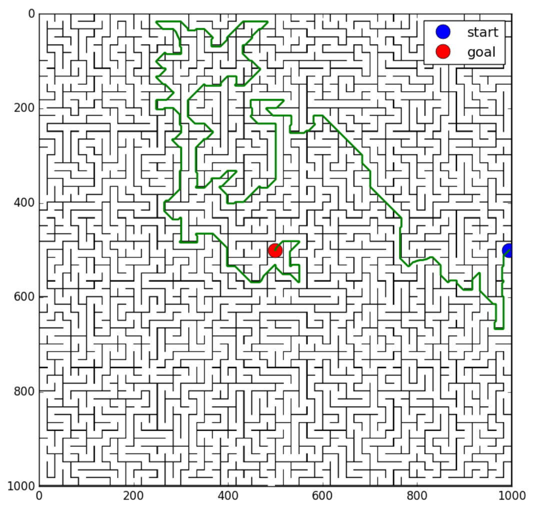


Fig. 33: Path planning can be used to find a collision-free path to a goal.

BUG etc.) [96, 97], creating potential fields and using gradient descent or fast-marching method [98, 99] among many others. However, there are two fields which are seeing more usage than others: graph-search and sampling-based methods. The work in this thesis has had the most inspiration from these two categories, and as such the focus will be on them rather than doing an extensive review of the whole field of path planning, which would be out of the scope and the reader is instead referred to books such as [100, 101].

3.1 Configuration-Space

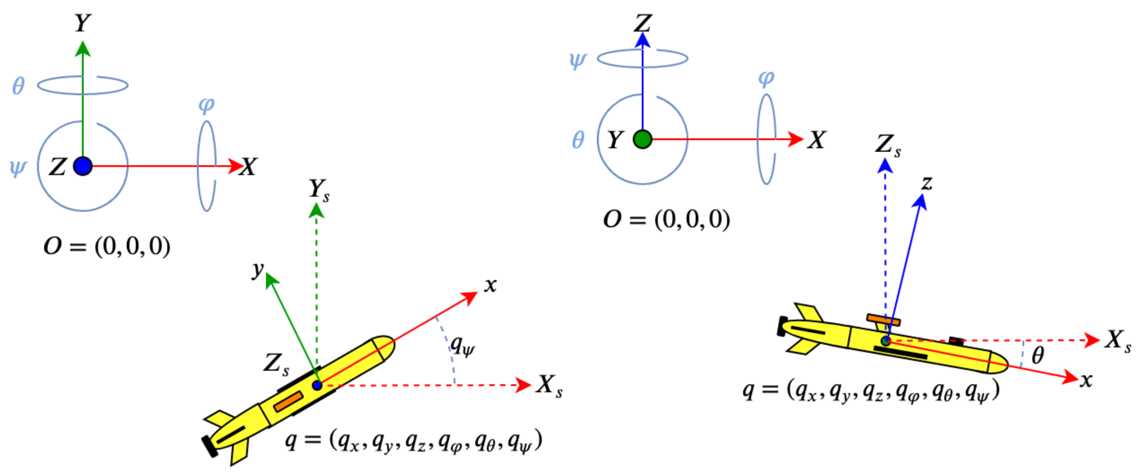
Before presenting the algorithms used for path-planning the representation of the robot and the world they operate in needs to be defined. This is often based on Lozano-Perez's definition of Configuration Space (C-Space) [102]. The C-Space represents the set of all possible configurations of the robot in the *workspace*. The workspace, $\mathcal{W}$, is the $\mathbb{R}^n$ where n is the dimension in which the robot operates in (usually 2 or 3). The state of a robot is represented as a *configuration*[1] (q). The configuration of the robot depends on its position and potentially also its orientation. For this thesis we only consider rigid body vehicles, and as such the pose of the robot is not necessary to describe, in contrast to for example a robotic arm with n joints having a post as a set of n configurations ($q = [q_1, .., q_n]$). Following the notation in [100], the following will be used throughout the thesis:

- $\mathcal{C}$ - The configurations space
- $\mathcal{W}$ The workspace, also noted as $\mathbb{R}^n$ for dimension n
- q a configuration in $\mathcal{C}$. q consists of the following:
  - Position in $\mathbb{R}^n$
    - $\mathbb{R}^2 \rightarrow$ [x(east), y(north)]^T
    - $\mathbb{R}^3 \rightarrow$ [x(east), y(north), z(depth)]^T
  - Orientation ($\mathcal{SO}(n)$) in $\mathbb{R}^n$
    - $\mathcal{SO}(2) \rightarrow$ [yaw(ψ)]^T
    - $\mathcal{SO}(3) \rightarrow$ [roll(φ), pitch(θ), yaw(ψ)]^T

    Hence giving $q \in \mathcal{C} = SE(n) = \mathbb{R}^n \times \mathcal{SO}(n)$ the following:
    - $q \in \mathbb{R}^2 = [x, y, \psi]^T$ (see Fig. 34a)
    - $q \in \mathbb{R}^3 = [x, y, z, \varphi, \theta, \psi]^T$ (see Fig. 34b)

With this, all configurations of the robot can be described. However, not all configurations are valid due to obstacles and limitations in the environment. An obstacle can be described as a configuration with a shape, as such we define an obstacle $\mathcal{O}_i$ in $\mathcal{C}$ as $\mathcal{CO}_i$. With all known obstacles ($\mathcal{O}_{i \forall i \in [1,..n]}$) we can define the free C-Space as $\mathcal{C}_{free} = \mathcal{C} \setminus \cup_i (\mathcal{CO}_i)$. We can now describe a valid configuration $q \in \mathcal{C}_{free}$, which henceforward a valid configuration q will be assumed to be in, unless otherwise stated. With the representation of the robot and the world defined we can now introduce different approaches to find a path within $\mathcal{C}_{free}$ for a robot to reach a desired configuration.

[1]Configuration and state in literature often describe the same thing.


Y
θ
φ
ψ
Z
X
O = (0,0,0)
Y_s
y
x
q_ψ
Z_s
X_s
q = (q_x, q_y, q_z, q_φ, q_θ, q_ψ)
Z
ψ
φ
θ
Y
X
O = (0,0,0)
Z_s
z
X_s
θ
x
q = (q_x, q_y, q_z, q_φ, q_θ, q_ψ)

(a) top view (X-Y plane) (b) Side view (X-Z plane)

Fig. 34: Configuration of a state. For a vehicle operating in a 2D C-Space (such as an ASV) (a) can be used and the state is then $q = [q_x, q_y, \theta]$

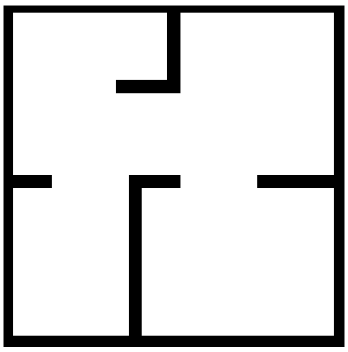
3.2 Search-Based Methods

Search-based methods, also called graph-based method, is an approach to finding a solution searching a graph. A graph consists of vertices[2] and edges. An edge (E) represents a relationship between two vertices, such as a feasible path for a robot to go from state q_a to state q_b (edge E_{ab}). An important aspect of an edge is if E_{ab} is feasible it does not mean that E_{ba} is. This is especially true for robots which cannot move freely in all dimensions in $\mathbb{R}^n$ such as a torpedo-shaped AUV, which needs to maintain a positive forward speed for control. In this case the edge is called *directed*. Other types of vehicles, such as a car, can reverse to go back to a previous state, making the edge *undirected*. A graph can either be pre-constructed (such as a map of roads between cities) or it can be created from the algorithm. From here on, we will consider creating the graphs to be created within the algorithm, as there are for marine robots no dedicated roads or maps determining how to move from one state to another.

[2]Also referred to as nodes, representing a robot's state or configuration.

3.2.1 Grid-Based

A grid-based approach to solve the path planning problem divides $\mathcal{C}_{free}$ into discrete cells. Each cell then represents a possible configuration for the robot ($q_{1,..,n} \in \mathcal{C}_{free}$). The cells can have various shapes and sizes, however most commonly they will be of equal size in the shape of a square or a cube depending on if the dimension is $\mathbb{R}^2$ or $\mathbb{R}^3$.


(a) Map

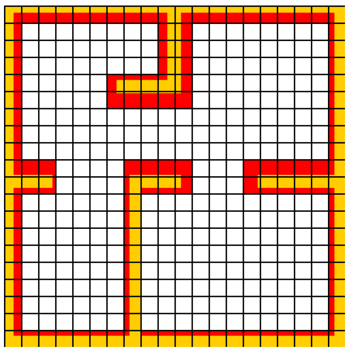
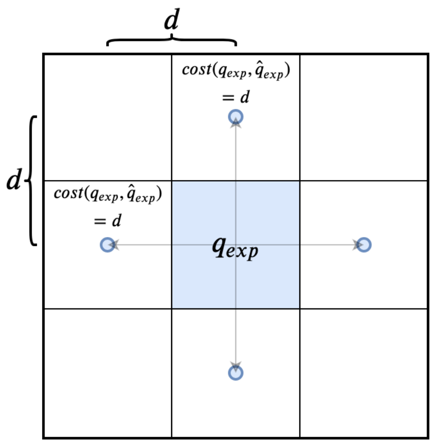

(b) Map with grid ($\mathcal{C}_{free}$)

Fig. 35: A map (a) can be divided into a grid (b) where the white cells are $\mathcal{C}_{free}$. The red cells have parts of them occupied by an obstacle and hence the whole cell is deemed as occupied.

With the C-Space constructed, the next step is to decide the motions allowed for the robot, this can be for example the closest neighbouring cells as seen in Fig. 36. Each movement is assigned cost, commonly to represent the length of the movement. However, various costs can be constructed based on what the goal of the planner is, such as energy optimisation[103] or to include ocean currents [104].


d
$cost(q_{exp}, \hat{q}_{exp}) = d$
d
$cost(q_{exp}, \hat{q}_{exp}) = d$
q_{exp}
(a) Motion allowed between neighbouring cells.

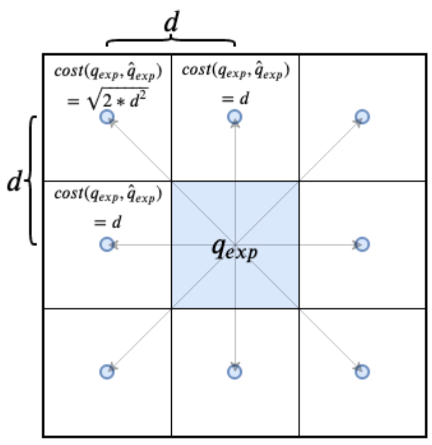

d
$cost(q_{exp}, \hat{q}_{exp}) = \sqrt{2 * d^2}$
$cost(q_{exp}, \hat{q}_{exp}) = d$
d
$cost(q_{exp}, \hat{q}_{exp}) = d$
q_{exp}
(b) Motion allowed between neighbouring and diagonal cells.

Fig. 36: A commonly used set of discrete motions used for to find a path in Dijkstra's and A* within a discrete grid

With this set of rules for possible motions and a discrete grid constructed, a planner can be used to find the path from the initial configuration X_{start} to the goal X_{goal} using various search methods. One of the first approaches to solve this was presented in the late 1950's — Dijkstra's Algorithm [105]. The algorithm, which can be seen in Alg. 1, is based on iteratively expanding the node (cell) with the lowest cost from a list of unexpanded nodes. The cost of a node is calculated by adding the cost of the expanding node and the cost for the movement (see equation (43)). This is performed until a node, which is in the cell containing X_{goal}, is opened for expansion. When this node is opened, a solution is found ($q_{solution}$). By retracing the path in which the cells were opened to reach $q_{solution}$, the shortest path can be found. The way Dijkstra's Algorithm searches makes it a breadth-first search algorithm.

$$g(q_i) = \sum_{j=2}^{i} motion\_cost(q_{j-1}, q_j) \tag{43}$$

The cost of a node in Dijkstra's algorithm is equal to the cost of the path to reach it from the root of the search tree, as seen in equation (43) ($f(q) = g(q)$).

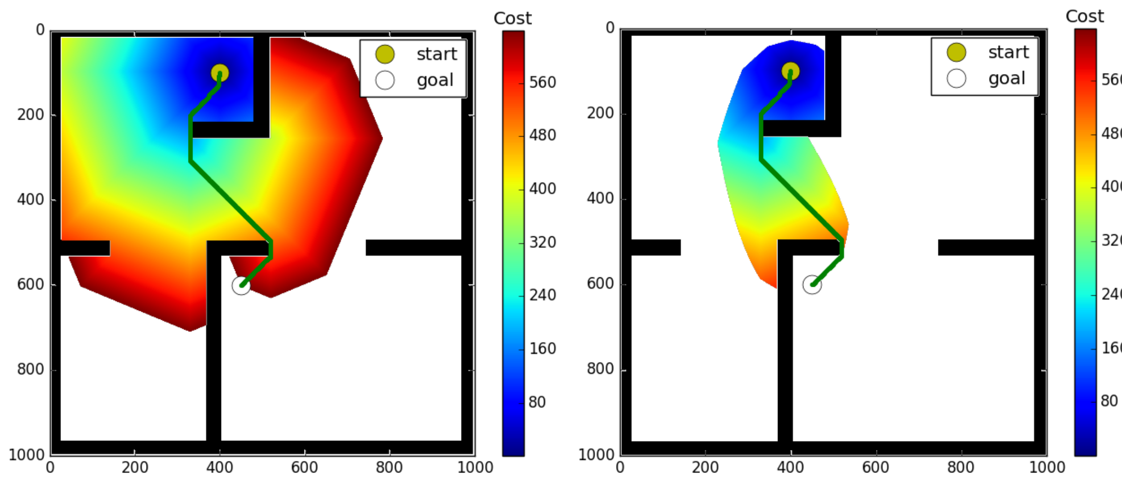
Algorithm 1 Dijkstra's Algorithm

```
Input:
$X_{start} X_{goal}$ : start and goal configuration
$\mathcal{Q}$ : cells $q \in \mathcal{C}_{free}$
 1: procedure DIJKSTRA'S
 2:     $grid = size(\mathcal{Q}) * \infty$
 3:     $q_{start} = X_{start}$
 4:     $grid(q_{start}) = f(q_{start})$
 5:     $OpenList = PriorityQueue()$
 6:     $OpenList.insert(f(q_{start}), q_{start})$
 7:     while $OpenList! = \emptyset$ do
 8:         $q_{exp} = OpenList.pop()$
 9:         if $q_{exp} == X_{goal}$ then
10:             return $q_{exp}$
11:         for each $\hat{q}_{exp} \in \mathcal{Q}_{Neighbours}(q_{exp})$ do
12:             if $f(\hat{q}_{exp}) < grid(\hat{q}_{exp})$ then
13:                 $grid(\hat{q}_{exp}) = f(\hat{q}_{exp})$
14:                 $OpenList.insert(f(\hat{q}_{exp}), \hat{q}_{exp})$
15:     return $\emptyset$
```

Roughly 10 years later, Hart *et al.* presented A* [106] which is an extension to Dijkstra's algorithm which instead of a breadth-first uses a guided search method. A* adds a heuristic cost to the function determining in what order to expand nodes. This heuristic is usually a cost based on the shortest path from the expanded configuration to the goal. The search then becomes informed, being able to find the same path as in Dijkstra's and doing so in worst case equally fast as Dijkstra's but in most cases in a significant reduced time.

To change Alg. 1 to incorporate the heuristics as in A* the cost for a node is changed to $f(q) = g(q)+h(q)$, where $h(q)$ is the heuristics which could for example be the Euclidean distance as in (44).

$$h(q) = |q_{goal} - q| \tag{44}$$



(a) Dijkstra's (b) A*

Fig. 37: A* finds a path with the same length as Dijkstra's but in less amount of explorations by using an informed search strategy by incorporating heuristics. Cells have the resolution of 1 unit (whole map is 1000×1000 units).

In Fig. 37 and table 5 a comparison between Dijkstra's Algorithm and A*, regarding iterations to find a solution and how long the path length of the solution is have been performed. A solution is considered to be found when a node is expanded within 50 distance units form X_{goal}. The path length of the solution is then considered to be the length found path + the distance to the centre of the goal region[3]. The comparison considers various cell sizes. It can be seen that A* finds paths of the same length as Dijkstra's Algorithm, while expanding less nodes and that the finer the resolution (smaller cell size) the shorter is the found path, but at the cost of longer execution time as more iterations are needed.

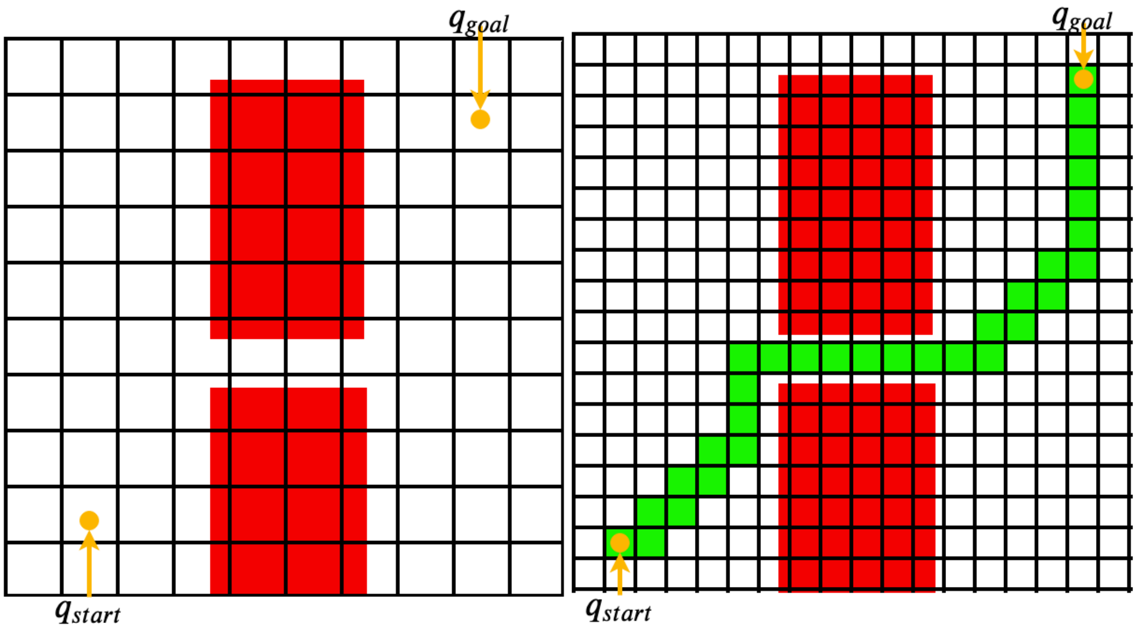
An important aspect of these planners is that they are *resolution complete*, meaning that they will find a solution (based on possible motions and the discrete grid) if one exists, or otherwise are able to report that no solution exists (Line 15 in Alg. 1).

The same problem might both be solvable and not solvable by grid based approaches depending on the size of the cells. A too large cell size can potentially lead to making a scenario un-solvable. An example of this can be seen in Fig. 38.

[3] The reason for using a region as a goal and the distance to the centre of the region is due to making a fair comparison to other algorithms which will be described in later sections of this chapter

	Cell size	Path length	Nodes
Dijkstra's	1.0	637.3	347196
	5.0	640.8	13904
	10.0	645.0	3496
	25.0	645.0	632
A*	1.0	637.3	139186
	5.0	640.8	5885
	10.0	645.0	1556
	25.0	645.0	289

Tab. 5: A comparison of Dijkstra's and A* in the same scenario as in 37 regarding the number of nodes explored and length of found path depending on the cell size/step length.



(a) Low resolution grid - no solution

(b) High resolution grid

Fig. 38: The size of the cells in discrete grids might contribute to whether a planning problem is solvable or not. It also determines how many iterations are needed to find a solution and the length of the solution.

3.3 Sampling-Based Methods

Another family of algorithms that have been seeing much usage is sampling-based methods. Sampling-based methods do not operate in the same discrete C-Space as the grid search algorithms does. Instead it allows all of $\mathcal{C}_{free}$ to be used as potential states. The general approach for sampling-based path planning methods is performed in two steps which are *sampling* and *connection*. The sampling step samples random point(s) in $\mathcal{C}_{free}$. The connection step connects the sampled point(s) to the current graph/tree. Sampling-based approaches have multiple benefits over grid-based ones, including:

- Constructing $\mathcal{C}_{free}$ for a discrete grid can be computationally expensive and grows exponentially with the number of dimensions and size.

- If a too large cell size is used in the grid, it might block the shortest path, or even all paths (which have been shown in Fig. 38). Allowing all of $\mathcal{C}_{free}$ to be used enables a collision free path to be found, if such exists and enough time for planning is given.

- The path is no longer limited by the discrete set of motions (example as seen in Fig. 36), hence having the possibility to find shorter paths.

Sampling-based methods however have a major drawback which is that they cannot guarantee to find a solution, even if one exists. Neither can they report that no solution exists in contrary to the grid-based methods (as can be seen at line 15 in Alg. 1, where an empty set is returned if no solution is found). Random sampling-based path planners are however *probabilistically complete*, meaning that the probability to find a path, if such exists, goes towards one if enough time/samples are given [107].

3.3.1 Probabilistic Roadmaps

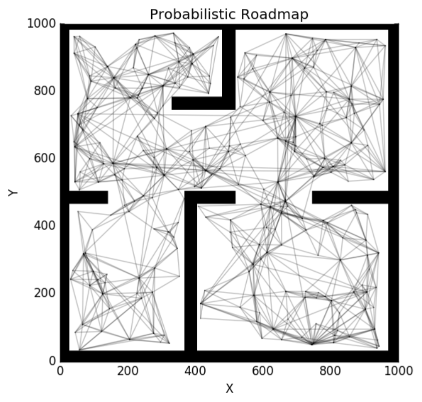
Kavraki *et al.* presented a random sampling approach to construct a roadmap [108] named Probabilistic Roadmaps (PRM). A roadmap is useful in scenarios where multiple queries to find a path within $\mathcal{C}_{free}$ is desired. It consists of a set of vertices connected to each other by edges. A vertex can be connected to multiple other vertices. The construction of the roadmap is performed using a *pre-processing* phase.

- In the *pre-processing* phase, the algorithm constructs a roadmap (through sampling- and connecting-phases) from n randomly sampled points ($q_{r[1,..,n]} \in \mathcal{C}_{free}$). Each point in q_r is then attempting to connect an edge to its m closest neighbours (if the edge is collision free). This constructs a map of n nodes, where each have attempted to connect to its $\leq m$ nearest neighbours. An example of different n and m can be seen in Fig. 39.

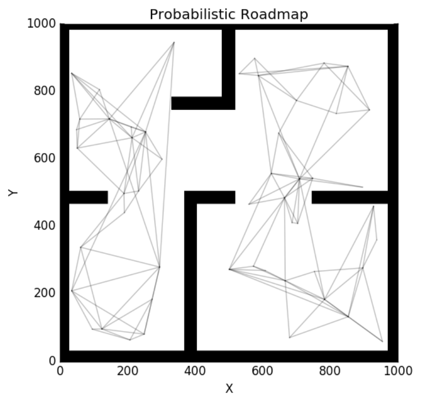
- With a roadmap constructed, a *query* to find a path from q_{start} to q_{goal} can then be performed. This step adds a q_{start} and q_{goal} state and connects these to the roadmap. The path between q_{start} and q_{goal} can then be found by graph-searching algorithms such as Dijkstra's or A*.

3.3.2 Rapid-exploring Random Trees

PRM requires an initially large computational effort to construct its roadmap. Whilst the roadmap can be re-used if the C-Space is static and there are no changes to $\mathcal{C}_{free}$, it might not be as efficient if new information is integrated in the map or if just a single query is



(a) 250 vertices with 5 edges



(b) 50 vertices with 3 edges

Fig. 39: PRM with two different number of vertices randomly sampled in $\mathcal{C}_{free}$ and different number of closest neighbours attempted to connect an edge to. If too few of either, it might result in multiple roadmaps which are not connected, as seen in (b).

desired. A single-query method using random sampling is Rapid-exploring Random Trees (RRT) [109], which was proposed by LaValle. The core of RRT is based on expanding a search tree from the root X_{start}, where each expansion is based on a random sampled point (q_{rand}) in $\mathcal{C}_{free}$ and expanding the closest node ($q_{closest}$) in the tree towards q_{rand} with an edge of a maximum specified length (see Max Distance in (c) in Fig. 40). If this edge is collision free; the edge and a new vertex is added to the tree. The expansion can be based on geometric or motion constraints. The method repeats the sampling and connection phase until either a solution is found or another termination condition such as maximum number of iterations have been performed.

The algorithm for RRT can be seen in Alg. 2 and visual explanation in Fig. 40. A simple approach to speed up the time to find a solution is to add a bias of selecting X_{goal} as the *random* point. This is usually done with a probability of $5 - 10\%$.

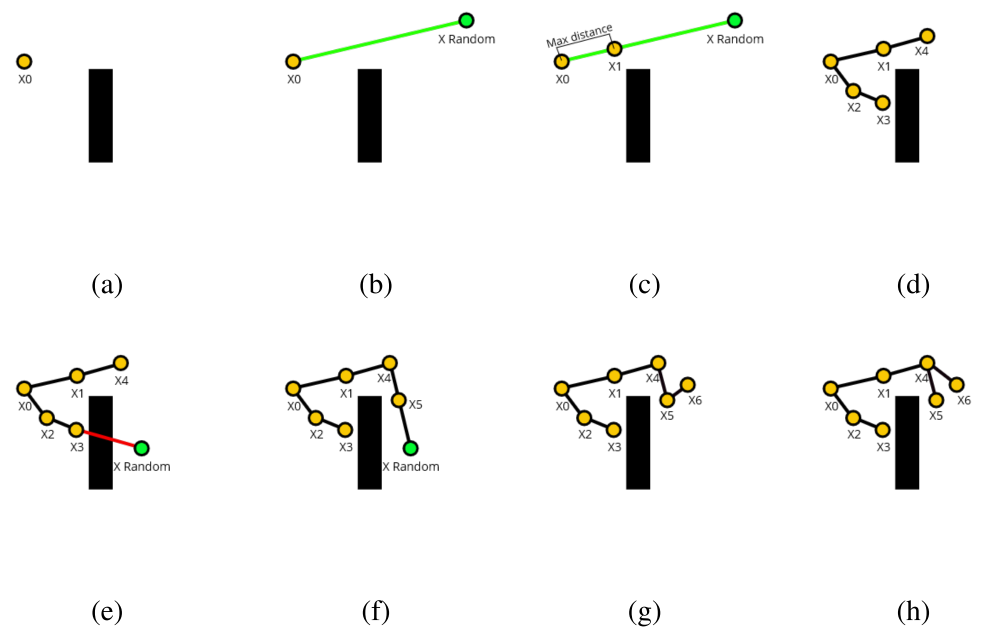


Fig. 40: Step-by-step visualisation of how an RRT is constructed, (h) shows how (g) would have connected X6 if RRT* would have been be used.

Algorithm 2 Rapid-exploring Random Trees

Input:
$X_{start} X_{goal}$: start and goal configuration
d : Threshold distance to accept as a solution
δ : expansion distance
Output:
A tree consisting of the vertices V, edges E and state considered a solution q_{exp}

```
1:  procedure RRT
2:      $V = [q_{start}]$
3:      $E = \emptyset$
4:      while $True$ do
5:          $q_{rand} = \mathcal{C}_{free}.random()$
6:          $q_{closest} = Closest(V, q_{rand})$
7:          $q_{exp}, collision = newExpansion(q_{closest}, q_{rand}, \delta)$
8:          if $!collision$ then
9:              $V.addNode(q_{exp})$
10:             $E.addEdge(q_{closest}, q_{exp})$
11:             if $|q_{goal} - q_{exp}| < d$ then
12:                 return $T(V, E, q_{exp})$
```

3.3.2.1 Asymptotic Optimal RRT (RRT*)

RRT is an algorithm good at exploring large spaces fast without taking the quality of the solution into account. This is as whenever a vertex is added and connected with an edge to the tree it does not change, meaning that the path will not be optimal nor can it be improved over time. In 2011, Karaman and Frazzoli presented RRT* [110] to handle this.

This is performed by adding a rewiring step after a new state (q_{new}) has been connected to the tree. The rewiring step selects a new parent for q_{new} (within a distance δ from q_{new}), if that would result in that the path cost to q_{root}/X_{start} would become shorter. It then checks the remaining nodes (within δ distance) to see if any of these could be connected to q_{new} to create a shorter path to q_{start}. An example of the rewiring process can be seen in Fig. 41.

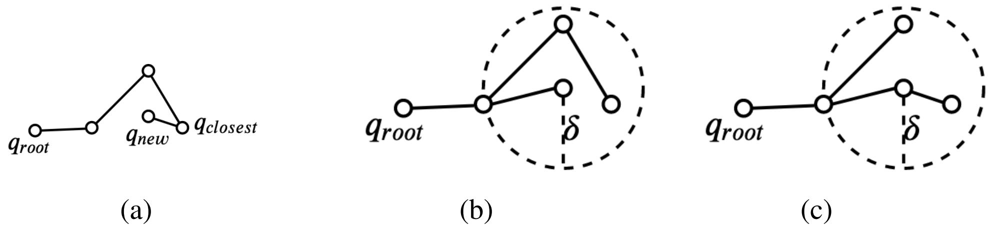


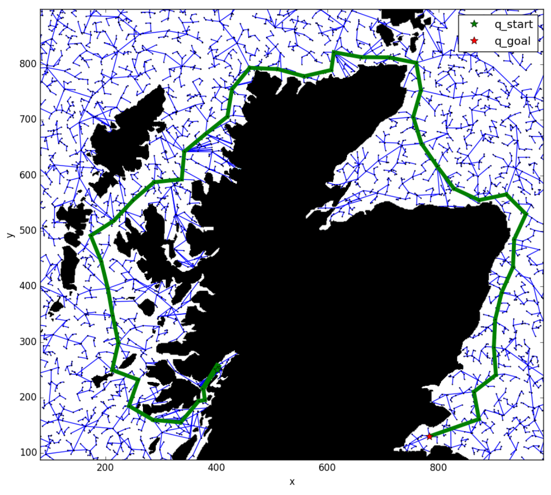
Fig. 41: The rewiring process. **(a)** A randomly sampled node q_{new} is connected to the closest node in the tree $q_{closest}$. **(b)** q_{new} is rewired to be connected to the node within δ distance the minimises the cost to q_{root}. **(c)** The nodes within δ distance from q_{new} calculates the cost to start if they were to change parent to q_{new}. If the distance is reduced they are rewired to connect to q_{new} instead.

How the exploration of RRT* compares to RRT on the same number of expansions which can be seen in Fig. 42. The asymptotic optimal property of RRT* comes from that the cost of a node, based on a user-defined cost function, can decrease with the number of samples [111]. As such, a better solution may be obtained by continuing to run the algorithm even after an initial solution has been found.

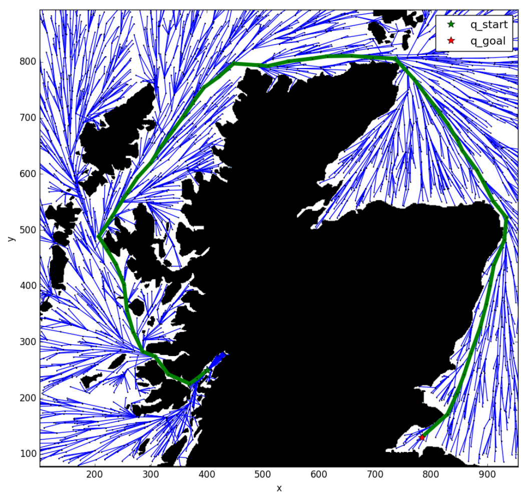
To speed up the convergence for improving the solution, the sample space can be reduce, as proposed by Gammell *et al.* in Informed RRT* (I-RRT*) [112]. I-RRT* uses RRT* until a first solution is found. When a solution has been found the sampling space instead becomes the union of $\mathcal{C}_{free}$ and an ellipse. The ellipse is based on the distance between q_{start} and q_{goal} and the current best solution. The ellipse has the property that no point outside it can improve the solution. This property is due to that if a point is selected on the perimeter of the ellipse and adding the from said point to q_{start} and q_{goal}, the cost will be equal to the current best solution. The sampling region can be seen in Fig. 43a, and an example of usage in Fig. 43b.

3.3.2.2 Comparison of RRT's

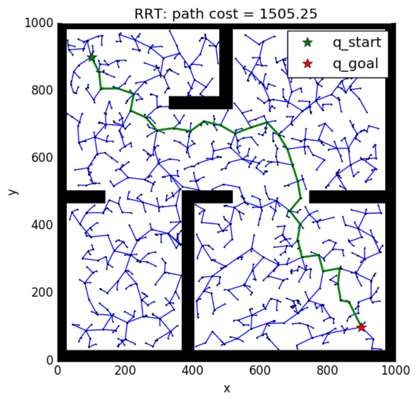
How RRT, RRT* and I-RRT* compare to each other in the same scenario as seen in Fig. 44, can be seen in Tab. 6. It can be seen that I-RRT* achieves the shortest path, and that the longer δ the faster the path is improved as it has a larger area to perform re-wiring in. For RRT* and I-RRT*, this produces a shorter path than A* and Dijkstra as can be seen in table 5.



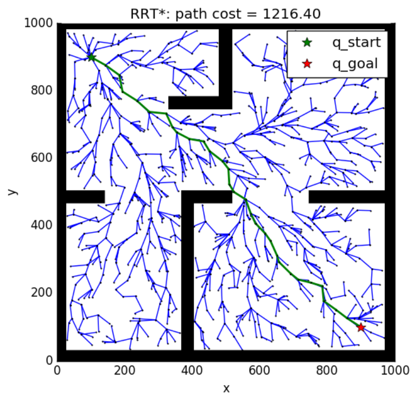
(a) RRT



(b) RRT*

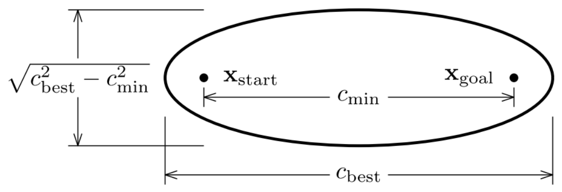


(c) RRT

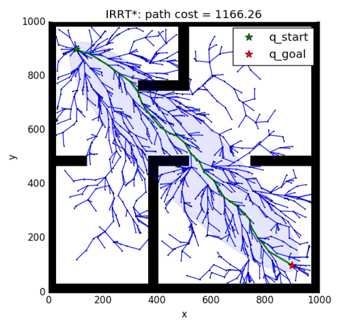


(d) RRT*

Fig. 42: Comparison between RRT and RRT*. (a) and (b) shows the result after 5000 iterations and (c) and (d) after 1000.

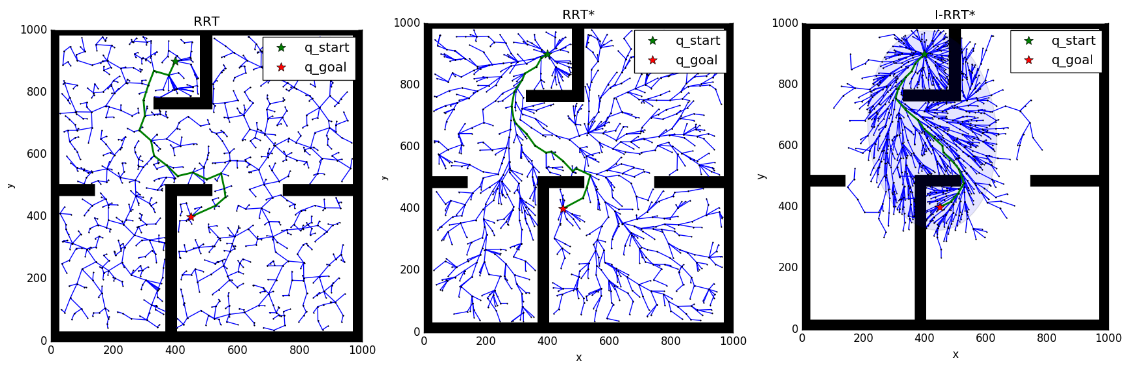

$\sqrt{c_{\text{best}}^2 - c_{\text{min}}^2}$
$\mathbf{x}_{\text{start}}$
$\mathbf{x}_{\text{goal}}$
c_{min}
c_{best}

(a) The region of sampling is based on the current shortest path and the Euclidean distance between q_{start} and q_{goal}.



(b) I-RRT*'s solution and sampling region after 1000 iterations.

Fig. 43: Once a first solution is found using RRT*, I-RRT* can improve the solution at a faster rate using an informed sampling strategy. This is based on only sampling nodes which have a possibility to improve the solution.



(a) RRT (b) RRT* (c) I-RRT*

Fig. 44: RRT, RRT* and I-RRT* after 1000 iterations.

	Step length	Max Iterations	Path length	First solution at [Iteration]
RRT	10	Until First	878.93	592.4
	25	Until First	1055.03	250.9
	50	Until First	919.37	125.1
RRT*	25	500	732.00	230.1
	50	500	702.83	103.2
	10	2500	959.50	720.0
	25	2500	685.43	275.9
	50	2500	633.05	131.5
I-RRT*	25	500	861.45	226.2
	50	500	688.92	112.9
	10	2500	962.65	786.1
	25	2500	658.03	295.0
	50	2500	624.57	129.2

Tab. 6: Average distance for solution found and iterations to find a first solution under different maximum step length and maximum number of iterations after 10 simulations for the same X_{start}, X_{goal} and map as in Fig. 44, the optimal solution has a length 616. This is the same scenario used for Dijkstra's Algorithm and A* in Fig. 37 and table 5.

3.4 Hybrid Methods

An approach to handle the highly discrete motion model used in Dijkstra's and A* was suggested by Ferguson and Stentz in Field-D* [113]. In their method, they allow linear paths between any points within a cell. This can create more smooth paths and results in a lower cost compared to A*. However, it does not handle the continuous motions which many vehicles have due to their kinematic constraints. One approach to handle motion constraints is HA*, which has been used for aerial vehicles [114] and autonomous cars [115, 116, 117]. Cars have a non-holonomic motion model and hence cannot use a motion pattern such as the one in Fig. 36. HA* is a hybrid method, it uses continuous motions and states within a discrete grid. In Fig. 45 a comparison between how the discrete grid is used between Dijkstra's/A*, Field-D* and HA* is shown.

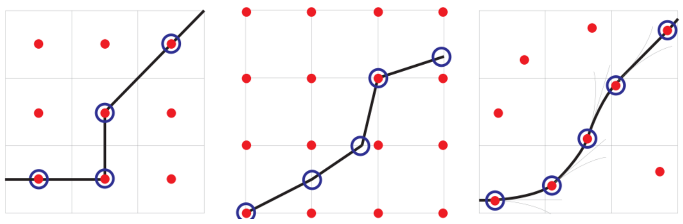


Fig. 45: Dijkstra's and A*'s states are stored as the centre of cells, Field-D*'s uses the corners of the grid and can connect an edge to any point at a edge between non-diagonal neighbouring corners. Hybrid-State A* (To the right) have continuous values within cells, allowing for a smooth continuous motion. Image-credit Dolgov *et al.* [116]

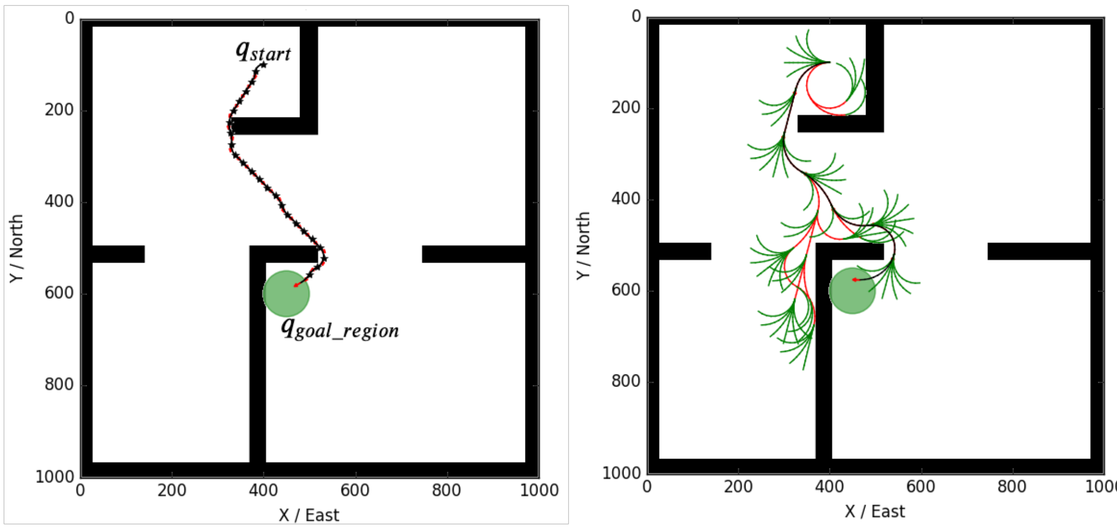
The big difference to previous grid-based methods is that in HA* a cell in the grid stores the state of the vehicle both as a continuous value in position and orientation. Priority of expansion of states is based on the same principle as in A* — the state with the lowest cost ($f() = g() + h()$) is selected for expansion. The expansion of a state is dependent on the configuration of the node being expanded and uses a motion model to calculate the new state's position and orientation. A new state cannot be created if it ends up in a cell which contains a state with a lower path cost ($g()$).

In chapter 4 HA* will be described in more detail as it has been extended from the usage for vehicles operating in a 2D plane such as autonomous cars to vehicles operating in 3D to be usable for torpedo-shaped AUVs. Torpedo-shaped AUVs are, like cars, non-holonomic but are able to change depth, hence the planning is in $SE(3)$ instead of $SE(2)$.

To show an example of a path planned by HA* for a vehicle with motion constraints in the same scenario as Dijkstra's/A* (Fig. 37) and RRT/RRT*/I-RRT* (Fig. 44) can be seen in Fig. 46. In the figure, the possible motions of the vehicle have been discretised into a set of possible turning radius. An example of how these can look like can be seen in Fig. 46b. The length of the motions along with the number of motions used for the expansion will, just as the cell-size and available expansion directions in A*, influence how many expansions are needed to find a path and the path cost of the found solution. HA* is *resolution complete* in the same sense as Dijkstra's and A*; if a path exists that can be reached with the specified motion capabilities, one will be found, and if it does not exist, the algorithm is able to report this.

3.5 Planning under Constraints

In the section for hybrid methods, HA* was presented which mentions planning a path based on the vehicle's motion constraints. This is for many fields of mobile robotics an important factor. If the goal is to find a path which should be followed as precisely as possible, e.g. to avoid obstacles, the path needs to be feasible for the considered robot. The typical motion patterns for A* and Dijkstra's in the grid seen in Fig. 36 is not achievable unless the robot is able to move without freely in all directions. This is however not the



(a) Small steps (length = 25 units)   (b) Large steps (length = 100 units)

Fig. 46: A path planned by HA* takes the kinematic model of the vehicle into consideration, ensuring a path which is both feasible (continuous motion) and collision free. In b) the red lines are states in the closed list (already expanded), green is currently in the open list (able to be expanded) and black is the found solution. The distance to reach the goal region+distance to the centre of the region is for (a): 600+47.32 and for (b): 700+29.48.

case for many robots. Instead the motion constraints of the vehicle should be taken into consideration while planning a path to ensure that it is feasible to execute. Another aspect that needs to be considered for online path planning is the time constraint to plan a path. A robot needs to be able to plan its path before it needs to be executed. For real-time operations where a vehicle cannot stop to plan its next trajectory, being able to plan a path within a strict time constraint is necessary. This section will discuss path planning from the point of planning under both motion and time constraints.

3.5.1 Real-time and Anytime Path Planning

While planning a path, the planning-time is often a tight constraint, since a robot needs to be able to plan a path before it needs to execute the plan. For some fields this time is not necessarily a constraint, such as a static robotic manipulator or an autonomous car in a static environment. This is an issue for other robots such as an aerial robot which cannot hover or a torpedo shaped AUV which needs to be in constant motion to have control over its movement. Such robots cannot wait for a plan to be finished, but need to have it solved within a limited time. Hence the need to be able to plan a path in real-time is of high importance.

For these types of scenarios, the ability to plan a path fast is a must, however, there is as of today no fast planning method that is consistent and optimal. As has been seen in previous path planners, some algorithms are able to calculate a non-optimal path fast.

Even if the path is not optimal, it is still a feasible start and a path that can be improved over time. This is where the term *anytime* in path planning can make a difference. Anytime path planning refers to the possibility to be able to improve a path by allowing the algorithm to run for a longer time. In this chapter, RRT* and I-RRT* are examples of where a first non-optimal solution is found, but by allowing more iterations the algorithms can keep improving the solution as can be seen in 47.

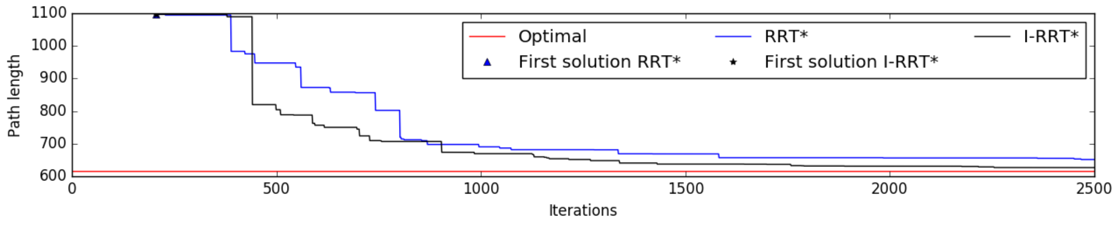


Fig. 47: RRT* and I-RRT* can improve the solution by additional execution-time after a first solution has been found. The scenario here is the same as in Fig. 44.

The path found by methods such as RRT* can be improved during execution. Karaman *et al.* uses RRT* to find a path. While the path is executing it continues to improve the path until the vehicle reaches its next planned state [118].

An approach to handle re-planning/reparation of paths created by A* is Dynamic A* (D*) by Stentz. It starts the search from the goal. D* is an incremental search method which can repair paths instead of re-planning from scratch [119]. This achieves a substantial speed-up. The algorithm was simplified by Koenig and Likhachec in D* Lite [120], which builds on the work of Lifelong Planning A* (LPA*) [121]. Lickhachev *et al.* have also presented an anytime version (Anytime Dynamic A* (AD*)) which incrementally improves the solution by updating the search by decreasing the weight of the heuristic cost function.

3.5.2 Planning under Motion Constraints

If one of the objectives for a path planner is to find a feasible and collision free path for robot, the motion constraints of the vehicle should be taken into account. Planning by taking the kinematic or differential constraints into consideration is also known as kinodynamic motion/path planning. An example where a path is planned based on the motion constraints of a non-holonomic robot were shown in Fig. 46b. It can be seen how all the explored paths have a smooth and continuous motion, which is able to be followed by a robot.

The set of motions described for Dijkstra's and A* have so far been assumed to be able to move as in Fig. 36. This is for many robots not a feasible set of motions as it assumes that they can move freely along all axes. An alternative approach can be seen in Fig. 48 [122]. This extension to the possible motions by Pivtoraiko *et al.* allows to one to apply the same algorithms (Dijkstra's/A*) but the expansion of a state is also dependent on the orientation of the vehicle. As such instead of operating in $\mathbb{R}^2$ a configuration is now in SE(2) (x, y and yaw as seen in section 3.1). It is important that the end of a motion ends in a discrete set of possible angles in the centre of a cell, as in Fig. 48. It can be seen that the vehicle is facing either north, east, west or south in each state. The number of discrete angles are however arbitrary as long as it is consistent. This extension can be incorporated by changing the $\mathcal{Q}_{Neighbours}(q_{exp})$ to return the set of possible expansion based on q_{exp}'s position and orientation, along with the cost of the new set of motions. This allows for planning of continuous paths in a discrete grid. The paths can also span over multiple cells allowing long continuous paths between two non-neighbouring cells [123].

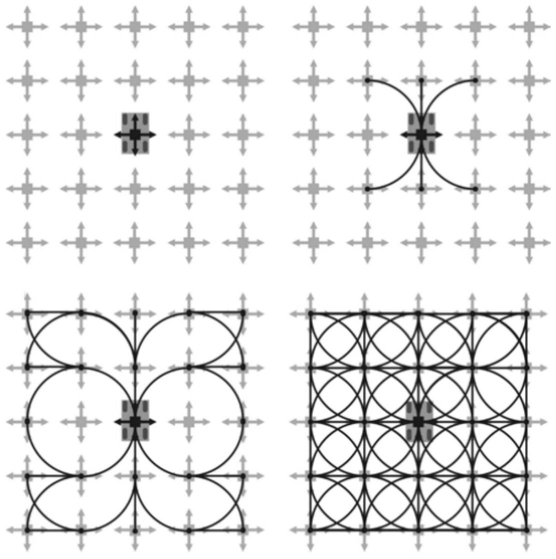
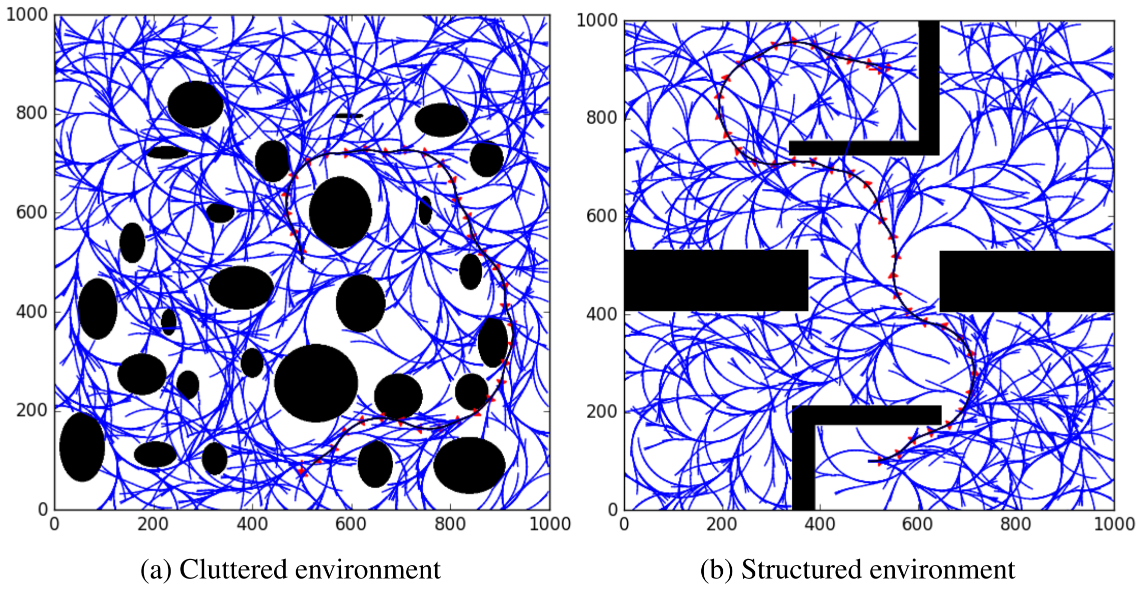


Fig. 48: Using a discrete set of motion capabilities, where each motion ends up in the centre of a cell allows for planning of continuous movements in a discrete grid. Image-credit Pivtoraiko *et al.* [122].

Using RRT to extend a tree based a vehicle with non-holonomic constraints was performed by Heo and Chung [124]. Their approach first builds a tree which is later solved by A* to find the shortest path to the goal region. An example of where RRT has been used taking motion constraints into consideration can be seen in Fig. 49.

Using a sampling-based approach which prioritises exploration of unexplored regions which are relevant to solve a query is performed in Expansive Space Trees (EST) [125, 126, 127]. It can extend the search tree based on the motion constraints of the vehicle. This was expanded to use a guided approach to reach the goal region in less explorations in [128]. An approach presented by Plaku *et al.* named Discrete Search Leading Continuous Exploration repeatedly finds a guided graph, a lead, which is then used as a guide to determine where to sample for expanding a search tree under the kinodynamic constraints. This is performed until a solution is found or a maximum execution time has been reached [129]. Vidal *et al.* also use a lead planned by RRT* to follow by a multi-layered path planner based on sparse-RRT [130].



(a) Cluttered environment

(b) Structured environment

Fig. 49: RRT is expanded using the specified motion constraints of the vehicle to ensure that the path is both collision free and drivable.

An extension to RRT* for constant depth planning for AUVs was proposed by Hernández *et al.* [131] where Dubins curves [132] is used to connect two states of vehicles based on **L**eft- and **R**ight-turns and **S**traight paths. This gives the following combinations of motions LSL, LSR, LRL, RSL, RSR and RLR, which can be used to connect any 2 configurations (considering a obstacle free space). An example of Dubins curves can be seen in Fig. 50. Dubins curves have also been used to plan paths in 3D for AUVs [133, 134].

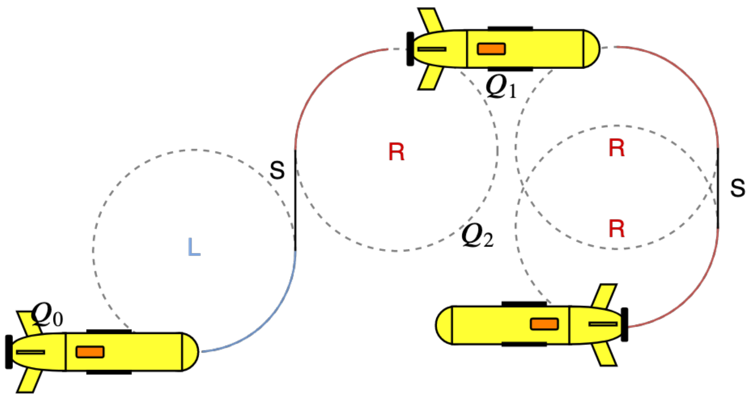


Fig. 50: Dubins curves performing a Left-Straight-Left action (from Q_0 to Q_1) followed by a Right-Straight-Right (from Q_1 to Q_2). Using Dubins curves with a minimum turning radius ensures that the planned path is feasible

Another approach to achieve drivable paths using RRT is to sample a set of random controls or by choosing from a set of controls and choosing the one resulting in the closest state to q_{rand}, as in Alg. 2 [135]. The interval of sampling can be limited by first estimating n different intervals using $n+1$ different turning commands (with the steering rate Ψ). For $n = 4$ the different steering rates could be $\Psi_{control_i} \in \{-\Psi_{max}, -\frac{\Psi_{max}}{2}, 0, \frac{\Psi_{max}}{2}, \Psi_{max}\}$ generating the n intervals as $\{[-\Psi_{max}, -\frac{\Psi_{max}}{2}], [-\frac{\Psi_{max}}{2}, 0], [0, \frac{\Psi_{max}}{2}], [\frac{\Psi_{max}}{2}, \Psi_{max}]\}$.

Choosing the $\Psi_{control_i}$ which generates the closest q_{near} and the neighbouring $\Psi_{control_{i\pm1}}$ closest to q_{rand}. The interval of sampling then becomes in the range between the closest and its best neighbour [136]. This can be seen in Fig. 51.

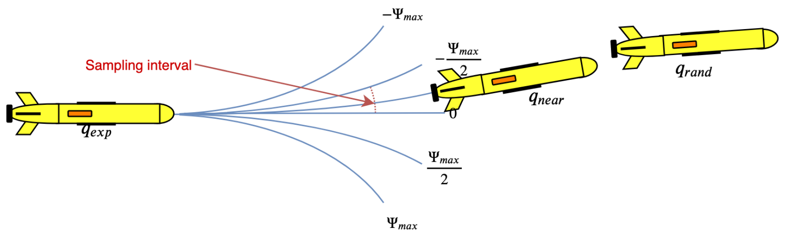


Fig. 51: Finding the interval which contains the closest states to q_{rand} to be within the steering rates $[-\frac{\Psi_{max}}{2}, 0]$. Random sampling a number of steering rates within this interval and selecting the q_{near} which is the closest to q_{rand} as the new expansion of the tree [136].

Another method which is used to plan feasible paths is the Dynamic Window (DW) approach is an online planning method which is based on sampling a region of reachable states. The region is dependent on the dynamics of the vehicle, and the sample which reduces a user-defined cost function (typically the one deemed to reach the goal the fastest) is selected as the next step for the robot to move towards [137]. The method can however get stuck in local minimas, and a global DW approach have been presented [138] which use a navigation function (based on wave-propagation from the goal) to calculate the distance in cells based on the distance to the goal. This is used to aid the dynamic window to calculate the cost to the goal when evaluating new states. It has been used for AUV in 2D [139] and 3D [140].

To ensure safe and feasible paths the motion model of the vehicle should be considered. However, the scale of operation is also important to take into account, in Fig. 42a and 42b a path from the Fort Williams, west coast of Scotland to Edinburgh, East coast of Scotland is found. At this scale taking the motion constraints might not be necessary for all vehicles, but instead it could be applied between waypoints along the found path. For large vehicles as tankers, this might however be of more importance as they can have a turning radius on the scale of kilometres.

3.6 Summary

This chapter has given an introduction and a review to some of the more common methods for path-planning along with some extensions to solve planning under time and motion constraints. Different algorithms have their own strength and weaknesses. The grid-based methods do not necessarily scale well with dimension nor in their standard form take motion constraints into account. The size of the cells within the grid can also cause a solvable problem to be un-solvable, or to make an easy problem very computationally expensive in the case of too small cell-size. They can however, always report if a solution exists or not depending on the resolutions which makes them a *resolution complete* planner. This is something that is not true for sampling-based methods. Sampling-based methods' probability to find a solution instead goes towards one as the number of iterations goes towards infinity, instead making them *probabilistically complete*. This *amount* of iterations is of course not possible, and for most problems the sampling based methods are efficient at finding a solution in any dimension relatively fast. These methods also have the potential to become improved over time by allowing more iterations. This property is called *anytime*.

It is clear that there is no *"optimal"* planner for all scenarios. This thesis will in later chapters present different path planners which are inspired by the ones mentioned in this chapter.

4 Online Path Planning in 3D Under Motion Constraints

For a path to be truly useful for an autonomous vehicle it needs to be planned while taking on the constraints and limitations of the vehicle into consideration. Methods such as Dijkstra's, A* and PRM do not necessary account for these. These could be kinematic and environmental constraints, and by using these during planning improve the possibility that a planned path is actually feasible for the robot. In section 3.5.2 some methods which have been used to plan paths for vehicles while taking the motion constraints of the vehicle into consideration are presented. Another approach which is mentioned in section 3.4, is Hybrid-State A*, which has proven successful for a non-holonomic vehicle such as a car [115, 116] which operates in $\mathcal{C} = SE(2) = \mathbb{R}^2 \times SO(2)$[1]. This chapter will focus on applying HA* to maritime vehicles. The vehicles considered are under similar constraints as a car, but are even more restricted in their motion capabilities. The first category is torpedo-shaped AUVs. These vehicles generally need a constant forward motion to be able to control their motions as they are naturally buoyant and would drift. Their motion capabilities includes turning and changing depth. Compared to a car they cannot reverse, and as such it is therefore important that a path planned is both feasible and collision free. If motion constraints are not taken into consideration a path might lead the vehicle into a state which is guaranteed to collide with an obstacle. The capacity to change depth makes them operate in $\mathcal{C} = SE(3) = \mathbb{R}^3 \times SO(3)$. The second type of vehicle that will be considered is an ASV which has the same motion capabilities as boats. They operate in the same C-Space as cars: $\mathcal{C} = SE(2) = \mathbb{R}^2 \times SO(2)$. However, as the dimension is higher for torpedo-shaped AUVs, planning for them is more complex and hence they will be the main focus of the chapter.

Torpedo-shaped AUVs are seeing much use in the maritime environment. They are primarily used for surveying big areas and exploration where energy efficiency is important. While the vehicles are often able to control their orientation in all 3 directions, this is not always desirable. The sensors they are equipped with are often used to sense the bottom. This can both be for mapping (side-scan sonars, cameras or lasers etc.) or to measure the velocity with a DVL. To achieve good data from these sensors it is important to have *bottom-lock*[2] This is equally true for missions under the ice where the sensors are

[1]See section 3.1

[2]If the sensors lose the sensing capabilities of the bottom they cannot produce reliable data for naviga-

directed upwards instead [141]. This chapter will take this into consideration and reduce the number of dimensions to plan in by removing the control of roll motions. Pitch motions can result in similar issues, but as this is needed for making a robot change depth it is still needed.

Other path planning methods have been used in the marine environment, most of these which take motion constraints into consideration have been sampling-based [124, 131, 133, 134, 130]. Sampling-based methods are however as stated in chapter 3, *probabilistically complete*, which means that they are not able to verify if no solution exists. Grid- or search-based methods can on the other hand verify if (based on grid- and cell-size and motion capabilities) a solution exists or not. The proposed version of HA* are also *resolution complete* and can hence report if a solution cannot be found.

The determining factor if a query is solvable or not is dependant on the motion capabilities and the environment. The presented algorithm in this chapter applies what is called a pattern to ensure drivable paths.

4.1 Motion Pattern

Planning a feasible path in 3D needs to be based on the motion capabilities of the vehicle. In this chapter the main focus will be on non-holonominc vehicles such as torpedo-shaped AUVs. These vehicles have a high limitation in their motion capabilites, which can be, for most vehicles of this class, be broken down to; forward-motions that includes turns and change of depths. This chapter presents what is called a *motion pattern*. This is a set of feasible motions that the vehicle of consideration is capable of. However, by defining other motion models of other types of vehicles such as a ROV, ASV or an AUV with higher degree of freedom in both dimension and orientation can be used by supplying a different motion pattern. A turning manoeuvre for a torpedo-shaped AUV can be calculated as a function dependent on the current configuration, the length of the desired trajectory with a radius around a point as seen in Fig. 52 [101].

A motion pattern consists of n branches (feasible motions). Each branch in its turn has m intermediate states. The intermediate states can be used for collision detection alongside checking for other constraints and limitations of the vehicle to evaluate if the branch is feasible. By adding multiple motion patterns, to turn both in the local X-Y plane and to change depth, a pattern such as seen in Fig. 53 can be obtained.

A motion pattern has its origin in a zero origin, where all coordinates and orientations in the branches are in relation to 0. To apply a motion pattern to an expanding state q_{exp} each of the intermediate states are transformed as in (46) using (45) in which ψ and θ is the pitch respective yaw orientation of q_{exp} along with its x, y and z position.

tion.

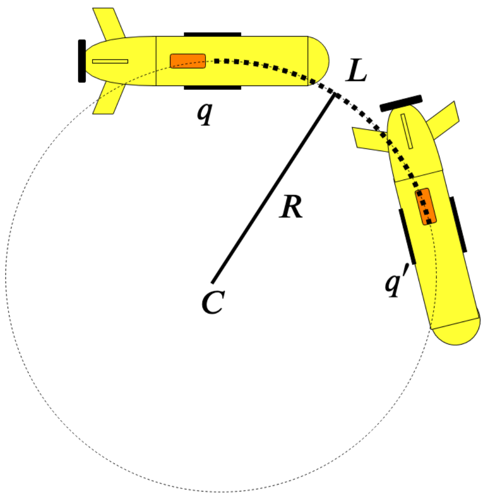
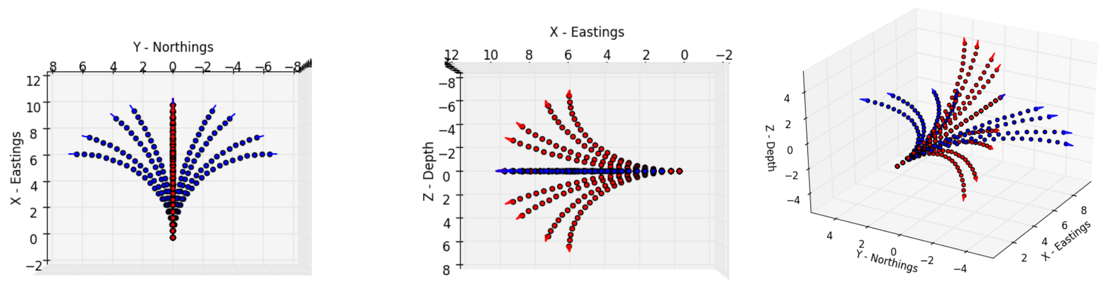


Fig. 52: The configuration q' can be calculated as a function dependent on q, the trajectory length L with the radius R around the coordinate C. This can be used to calculate a feasible motion (a branch in the motion pattern).



(a) X-Y plane (b) X-Z plane (c) Isometric view

Fig. 53: A motion pattern where yaw and pitch is controlled with 4 steering rate (in each of left, right, up and down) and moving straight forward allows for movement in 3 Dimensions.

$$T(x, y, z, \psi, \theta) = \begin{bmatrix} \cos(\psi) * \cos(\theta) & -\sin(\psi) & \cos(\psi) * \sin(\theta) & x \\ \sin(\theta) * \cos(\psi) & \cos(\psi) & \sin(\theta) * \sin(\psi) & y \\ -\sin(\theta) & -\sin(\psi) & \cos(\psi) * \sin(\theta) & z \end{bmatrix} \tag{45}$$

In (46), P is a branch's (Φ) intermediate state's (ϕ) configuration's position as: $[\phi_x, \phi_y, \phi_z]$.

$$q_{new} = T(x, y, z, \psi, \theta) * P^{\top} \tag{46}$$

After the position has been translated, the orientation is added as: $q_{new}^{orientation} = q_{exp}^{orientation} + [\phi_{psi}, \phi_{\theta}]$.

4.2 Pattern-Based Hybrid-State A* in 3D

By using a discrete set of motions for a vehicle to a create pattern as in Fig. 53, grid- and graph-based algorithms can be used. The end states of such motions will however not necessarily end in the centre of a cell or the edge between two corners of a cell. As such the representations of a state used in A* or field-D* (see Fig. 48) cannot be used, neither does this consider orientation of a configuration. A state in HA* is instead stored within a cell with the configuration's continuous value in position and orientation. A cell can also be divided into segments representing the orientation of a configuration. While the discritised version of¸in A* uses $\mathcal{C}_{free}$ where only a cell which does not have an obstacle in is a part of, Pattern-Based Hybrid-State A* (PBHA*) is able to use all of $\mathcal{C}$. This is as a configuration represents a position within a cell, but a cell can be partly occupied by an obstacle. Hence, as a too large cell-size which in the gird of A* could block a narrow passage (see Fig. 38b) this is not an issue using HA*. A cell is instead used to deny opening multiple expansions in the same cell if a new one has a longer path cost $(g(q))$ than the one already occupying the cell. Having a map from $\mathcal{C}$ along with the obstacles $\mathcal{O}_{i,..n}$, a motion pattern Ψ along with an initial configuration X_{start} and a goal X_{goal}, PBHA* can be used and is seen in Alg. 3.

Algorithm 3 Pattern-Based Hybrid-State A*

```
Input:
$X_{start} X_{goal}$ : start and goal configuration
$\mathcal{Q}$ : cells $q \in \mathcal{C}$
$\mathcal{O}$ : Obstacles
$\Phi$ : Motion pattern
 1: procedure PBHA*
 2:     $grid = size(\mathcal{Q}) * \infty$
 3:     $q_{start} = X_{start}$
 4:     $grid(q_{start}) = f(q_{start}, q_{goal})$
 5:     $OpenList = PriorityQueue()$
 6:     $OpenList.insert(grid(q_{start}), q_{start})$
 7:     while $OpenList! = \emptyset$ do
 8:         $q_{exp} = OpenList.pop()$
 9:         if $q_{exp} \in X_{goal}^{region}$ then
10:             return $q_{exp}$
11:         for each $\phi \in \Phi$ do
12:             $\hat{q}_{new}, Valid = expand(q_{exp}, \phi, \mathcal{O})$
13:             if Valid and $\hat{q}_{new}^{f} < grid(\hat{q}_{new})$ then
14:                 $grid(\hat{q}_{new}) = f(\hat{q}_{new}, q_{goal})$
15:                 $OpenList.insert(grid(\hat{q}_{new}), \hat{q}_{new}))$
16:     return $\emptyset$
```

In Alg. 3 at line 12, the code for how to expand a branch from a state (q_{exp}) is described in Alg. 4. In the expansion step, the intermediate nodes are checked for collision. If a collision is detected, the expand function returns an empty set and an indication that the expansion could not be completed, and hence the motion is invalid and should not be added to the open list.

Algorithm 4 Expand

```
Input:
$q_{exp}$ : State to expand from
$\phi$ : Branch of motion pattern
$\mathcal{O}$ : Obstacles
1: procedure EXPAND
2:     for each $\phi_i \in \phi$ do
3:         $q_{new}^{intermediate_i} = T(q_{exp})\phi_i$
4:         if $(q_{new}^{intermediate_i}) \in \mathcal{O}$ then
5:             return $\emptyset$, False
6:     return $q_{new}$, True
```

The expansion algorithm can be modified to incorporate various limitations such as maximum pitch angle or depth of the vehicle by extending the if statement on line 4 to handle this.

4.2.1 Implementation of Pattern-Based Hybrid-State A* in 3D

The HA* implementation described here is using a supplied pattern is done in the programming language C++ using Robotic Operating System (ROS)[142]. The benefits of extending HA* to be based on a supplied (pre-calculated) pattern is trifold: 1) The possible motions only have to be calculated once, reducing computational time. 2) Increasing the resolution is simple by adding more branches to the pattern. 3) The algorithm can be used on an arbitrary vehicle as all that is needed to change vehicle type is to supply a new motion pattern.

4.2.1.1 Collision Detection

To plan a collision free path, the objects in the known environment must be described in some way. In our implementation this is done with an Octomap [143]. By using a using a geometric model (such as a cylinder, box) or an actual model of the vehicle, a collision detection is performed by placing the model within the Octomap and seeing if it overlaps with any obstacles. This check is performed using Flexible Collision Library (FCL) [144]. Finding a path while using a precise model of the vehicle might end up in collisions when executed due to the control not being perfect and external sources of error and noise in the map. A method that can reduce risk from collision while executing a planned path can either be to have the obstacles in the map or the vehicle enlarged [100]. Another approach to reduce collision risk is to add risk zones around the vehicle [145] which adds a penalty to paths depending on which risk zone is overlapping with known objects.

The implementation described uses an incremental collision check [100], as seen in Fig. 54, by traversing each branch's intermediate state from the first to the last until either the last one is found to be collision free to return it as a possible extension. If it is a viable path, and the cell for q_{new} is either free or currently occupied with a higher cost configuration, q_{new} is added to the tree. If it is not valid, or the cell which q_{new} ends in is occupied by a lower cost state; q_{new} is discarded.

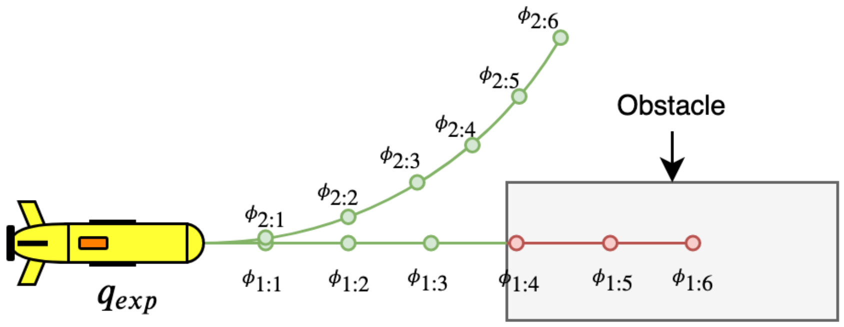


Fig. 54: Incremental collision check is performed by traversing the intermediate states of a branch in the motion pattern until either the end state is reached ($\psi_{2:6}$) or a collision is found ($\psi_{1:4}$).

4.2.2 Priority for Expansion of a Configuration

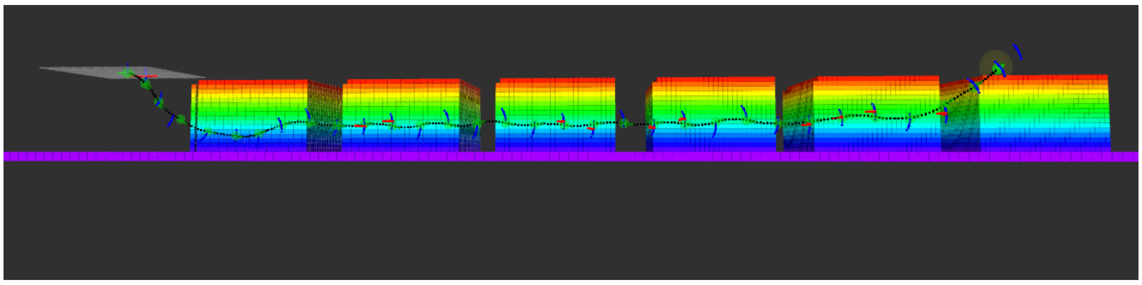
The priority for PBHA* to expand un-expanded configurations is based on a cost function. The cost function is, similar to A*, guided by an heuristic. A benefit of heuristic guided motion planners is that they prioritise expansion of states that are estimated to be closer to the goal. The cost $(f(q))$ for a state is considered to be the the path cost $(g(q))$ added with the heuristic cost $(h(g))$, as in (47).

$$f(q) = g(q) + h(q) \tag{47}$$

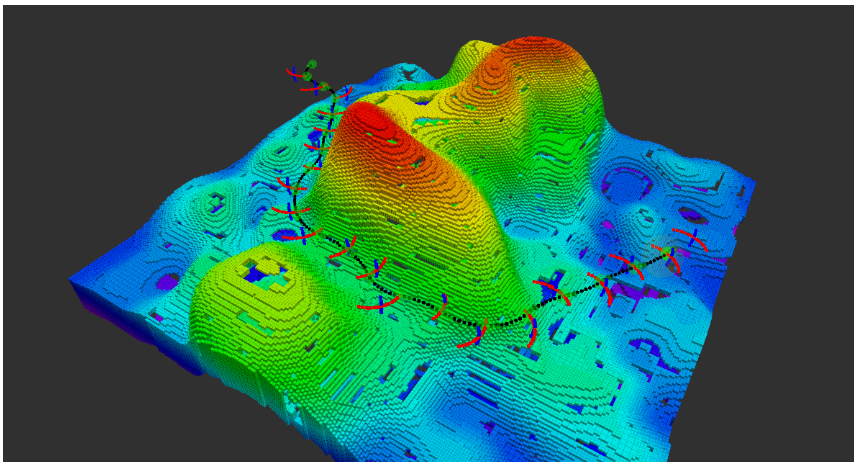
4.2.2.1 Path cost

The path cost, $g(q)$, for a state q_i as in equation (43), is commonly considered to be the length of the path from q_{start} to the q_i.

However, the cost can be weighted as, $branch\_length(q_{j-1}, q_j) * \epsilon_i$, for some application such as using a cost-map or a weight-function, similar to Transition-based RRT [146] to keep a specified depth or altitude among others. An example of such paths can be seen in Fig. 55.



(a) Adding a penalty to the path cost depending on how the estimated altitude would differ from a reference can make the planner aiming to plan to keep the vehicle at a certain altitude. This could be desirable for side scan data etc.



(b) By adding a penalty to the cost of a path for changing depth a planner which aims to find a feasible path which is in the plane can be obtained.

Fig. 55: By adding a penalty to the path cost ($g(q)$) the path can be more dependent on the known environment.

4.2.2.2 Heuristic function

The heuristic function ($h(q)$) represents what is estimated to be the minimum cost from one configuration to the goal. The heuristic cost function can be used to guide the expansion towards expanding states which are more likely to find a solution faster. That is by putting a bias towards states closer to the goal. If no heuristic is used, the search becomes breadth-first, which is what is performed in Dijkstra's algorithm. If the heuristic is equal or lower than the actual to the cost of moving from a state to the goal, it will find the optimal path based on the used pattern resolution. If the heuristic is instead higher than the cost to reach the goal the algorithm becomes a greedy, best-first search giving a bias to states closer to the goal. This is called a weighted heuristic. A typical heuristic cost function could be the Euclidean distance between a configuration and the goal, as in (44). However, using a weighted version of (44) as in (48) can speed up the algorithm while achieving at worst a solution which is maximum ϵ times longer [147]. The higher the ϵ, the more of a greedy the search becomes.

$$h(q) = \epsilon * ||q_{goal} - q|| \tag{48}$$

A comparison on how different weights affect the solution in terms of number of expansions, execution time and path length will be presented in section 4.3.2.1 and table 7.

4.3 PBHA* in Known 3D Environment

The first type of environments the algorithm is tested is in a completely *a priori* known static environment. The vehicle considered is a torpedo-shaped AUV. For the considered vehicle a pattern where each branch is at the length of 3 metres, divided into 10 equally spaced intermediate states is used. As the length of a branch is 3 metres, the resolution of the found path will also be in increments of 3 metres. For the presented 3D scenarios, a comparison of the found paths along with the time and expansions to find it can be seen in table 7.

4.3.1 Scenario 1: Blocks

The first considered scenario is based on a map from *Sant Feliu de Guíxols* in Spain (satellite view in Fig. 56). It consists of multiple blocks with the size $14.5m * 12m$ which are separated by 4 metres.

The vehicle's initial state is on the surface and the goal is to go to a region on the other side of the blocks which is 6 meters below the surface, and 30 respective 80 metres away in the X-Y plane ($X_{start} = [0, 0, 0]$ and $X_{goal} = [30, -80, -6]$). The goal-region is considered to be all points within 3 metres of X_{goal}. The scene is represented by an OctoMap [143], which can be seen along a planned path by PBHA* in Fig. 57.

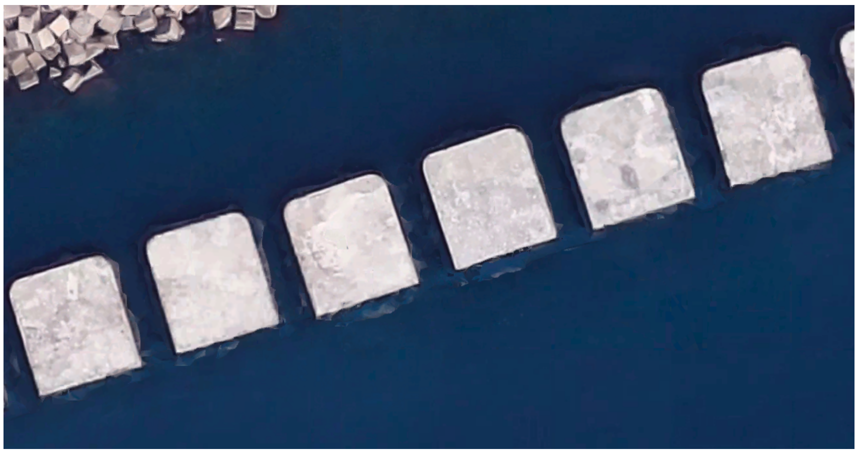
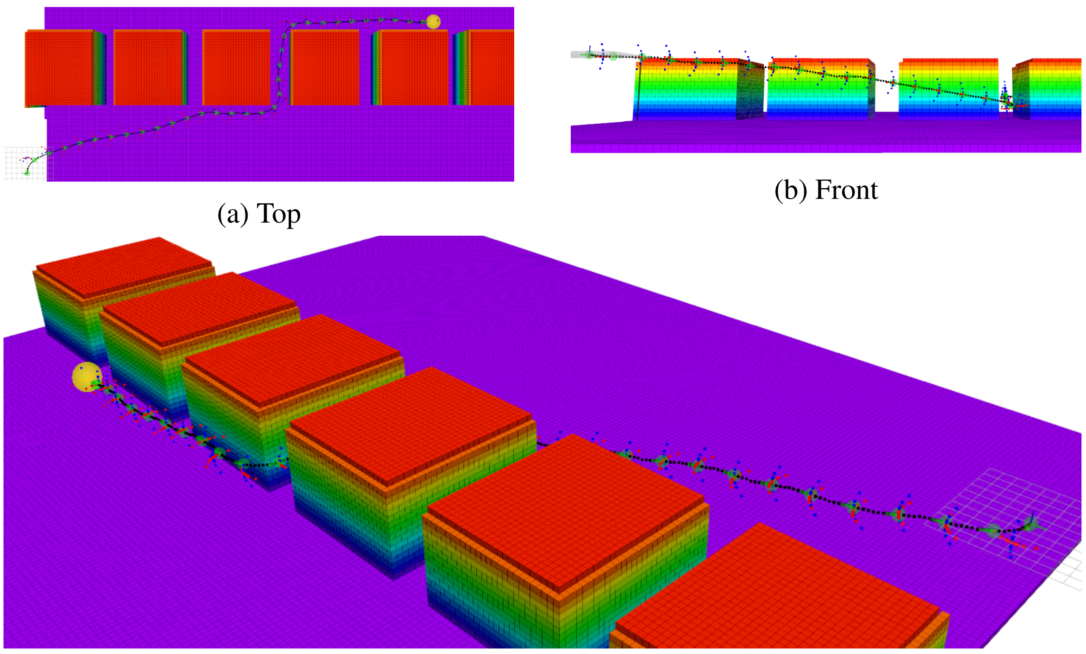


Fig. 56: Satellite image of the location where scenario 1 for PBHA* is based form. Image-credit Google.


(a) Top

(b) Front

(c) Isometric

Fig. 57: A path planned by PBHA* in a known environment. Black dots are the planned path and the green arrows are the orientation for the vehicle along the planned path (pitch and yaw). Blue and red dots are the end state of branches in the pattern along the travelled path's states.

4.3.2 Scenario 2: Underwater Terrain

The second scenario is in a simulated natural feature of topographical changes, which can be seen in Fig. 58. The vehicle's objective is to go from $X_{start} = [-5, 15, -6]$ to the region within 3 metres from $X_{goal} = [27, 45, -8]$.

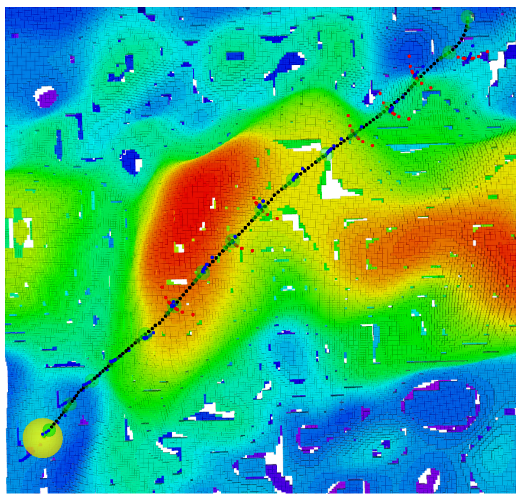
4.3.2.1 Weighted Heuristic Comparison

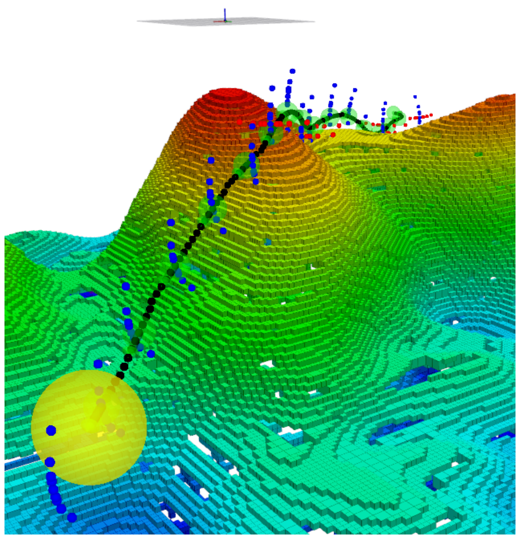
The selection of the weight for a weighted heuristic function greatly affect in which order the states are expanded. In table 7 a comparison of the two presented 3D scenarios can be seen.

In table 7, it can clearly be seen that a non-weighted (where $\epsilon = 1.0$) version of the Euclidean distance as heuristics achieves the shortest path. However, the planning time is high, making it unsuitable for online planning. If instead a weighted heuristic is considered, the planning time can be greatly reduced. In the blocks scenario, when high weight is used, the planning time is greatly reduced while the path length might be slightly increased. For the highest weighted scenario ($\epsilon = 3.0$), the planning time is reduced to roughly 0.2% of the time for the non-weighted version while only increasing the path length by 9%.

Weighted Euclidean Distance			
weight (ϵ)	Length [m]	Expansions	Time [s]
Scenario 1: Blocks			
1.0	99	6253	51.5721
1.1	105	4438	26.1212
1.2	105	2088	8.1494
1.3	105	553	1.4044
1.4	105	251	0.6402
1.43	105	203	0.4768
1.47	108	104	0.2305
1.5	108	68	0.1528
2.0	108	55	0.1202
3.0	108	43	0.1002
Scenario 2: Natural environment			
1.0	45	244	0.383
1.1	45	19	0.030
1.2	45	18	0.026
1.3	45	17	0.024
1.4	45	17	0.024
1.5	45	17	0.024
1.75	48	21	0.028
2.0	48	22	0.029
3.0	51	18	0.025

Tab. 7: Comparison of different ϵ values for weighted Euclidean distance as heuristic.


(a) Top


(b) Along path

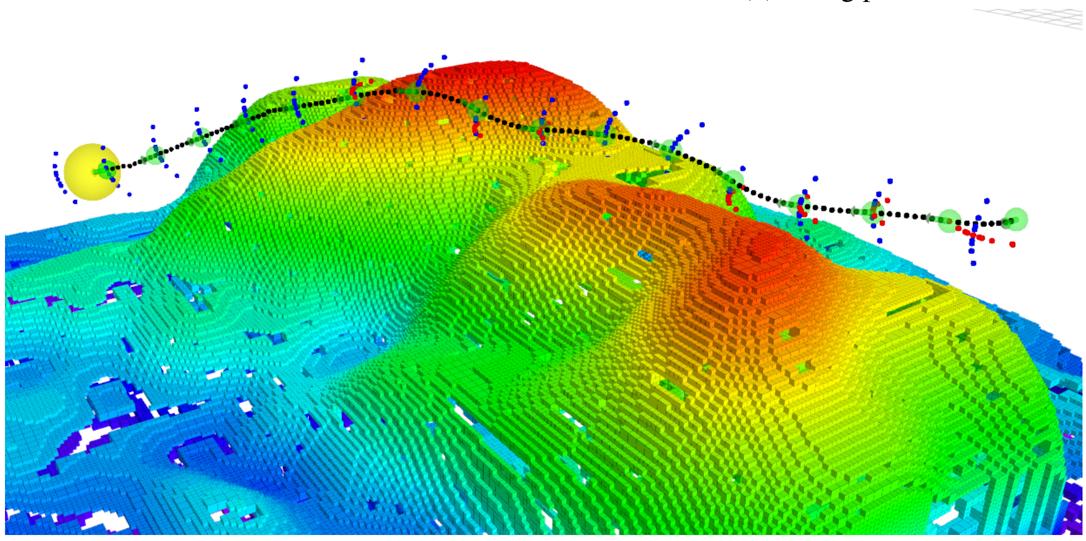

(c) Isometric

Fig. 58: a path planned by PBHA* in a known environment. Black dots are the planned path and the green arrows are the orientation for the vehicle in the planned path (pitch and yaw). Blue and red dots are the end state of branches in the pattern along the travelled path's states.

4.3.2.2 RRT comparison

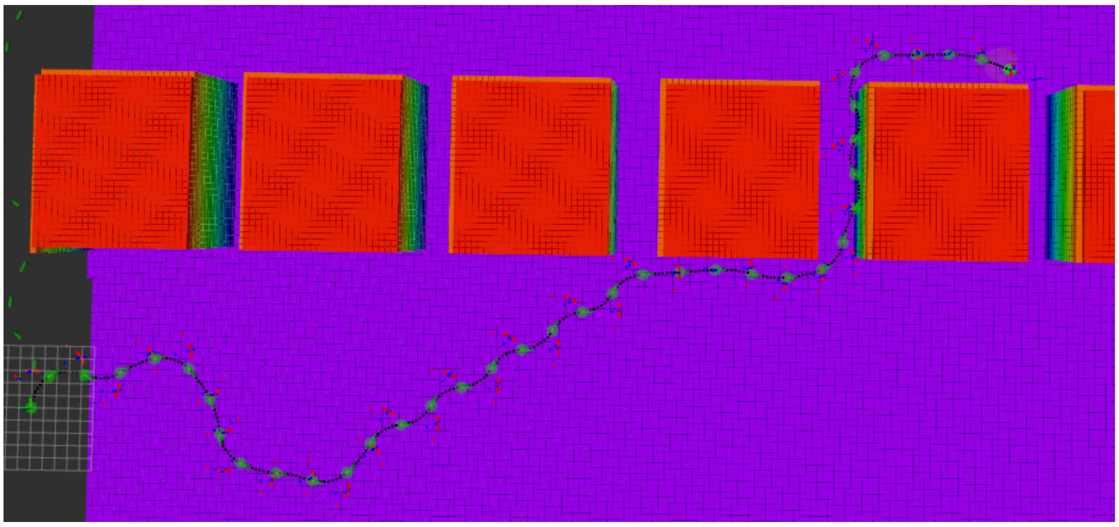
RRT has become a widely used algorithm for path planning showing success in various scenarios and dimensions as stated in section 3.3.2. A comparison between PBHA* to a version of RRT which uses the same motion pattern during expansion of a state has been performed. While RRT does not find optimal paths, it can give an indication of how it performs for finding an initial solution. As RRT is a stochastic method, 10 simulations of RRT have been performed for the same scenario as in section 4.3.1. The results can be seen in table 8. In Fig. 59 and 60, paths found from different simulations using the described RRT version are shown.

As seen in table 6 a comparison between RRT, RRT* and I-RRT* shows that RRT

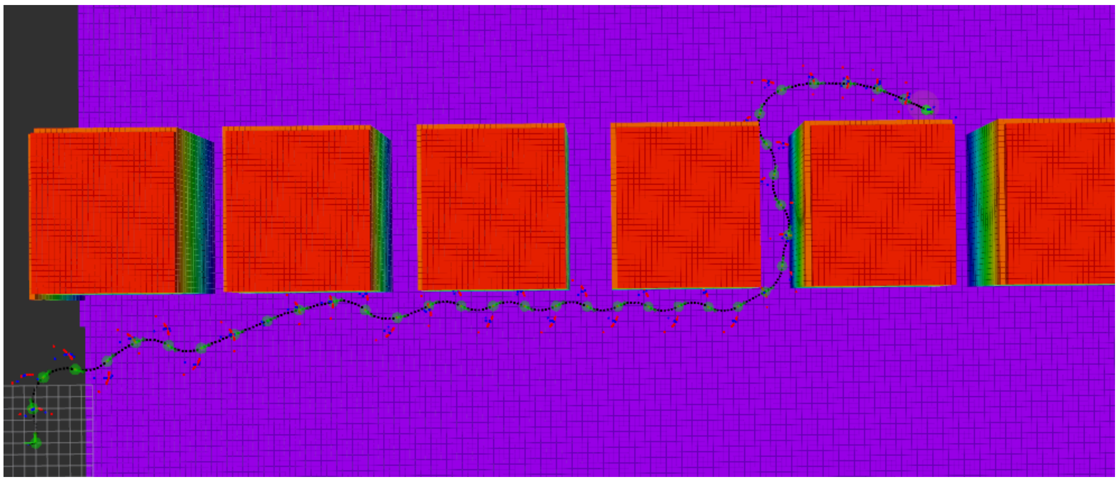
Scenario: Blocks				
Parameter	min	avg	median	max
Time [s]	0.2589	1.01156	0.4554	5.583
Expanded Nodes	156	479.9	284.5	2028
Path [m]	117	131.6	133.5	147
Scenario: Terrain				
Time [s]	0.117	7.55	2.00	25.97
Expanded Nodes	129.0	2240.29	1489.0	5977.0
Path [M]	57	75.86	75.0	96

Tab. 8: RRT using the same pattern, start and goal configuration as in Fig. 57 and Table 7. The results presented are based on the RRT method running 10 times.

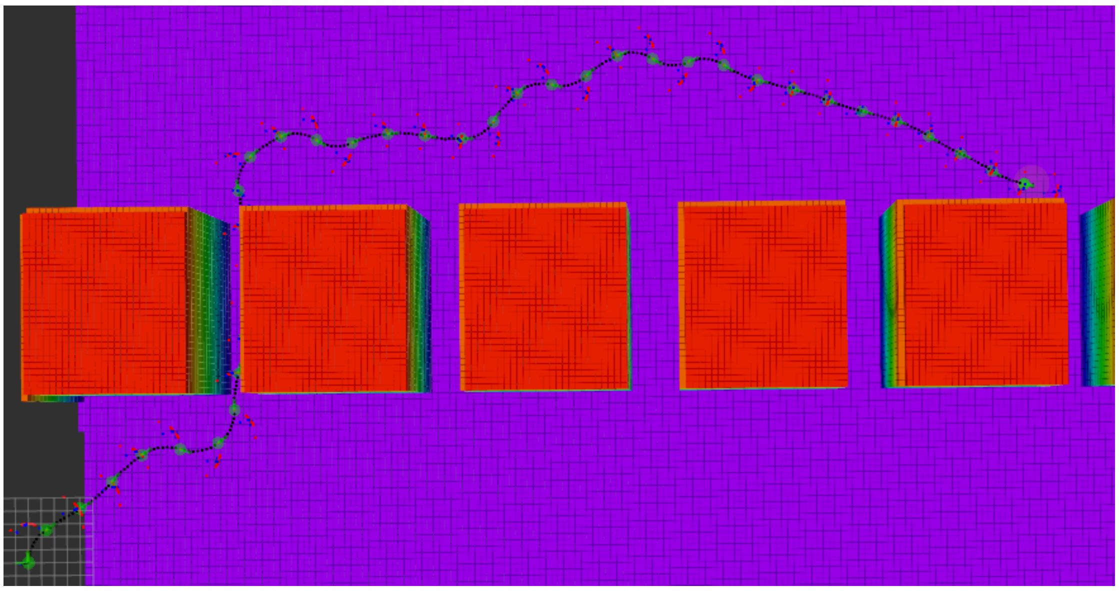
finds a first solution faster than the others. This is of interest when comparing PBHA* to RRT when they use the same method of expansion. When comparing the lowest values for RRT in table 8 to PBHA* in table 7 it can be seen that PBHA* performs better regarding execution times/number of expansion (except for $\epsilon = 1.0$), and achieves a lower path length in all scenarios compared to the best for RRT.



(a)

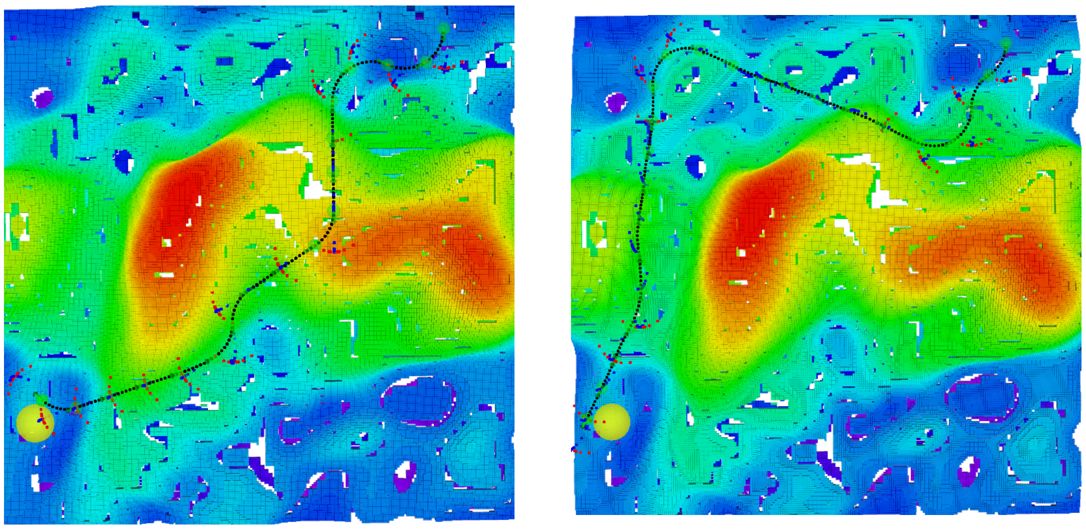


(b)



(c)

Fig. 59: Paths planned by RRT in a known environment. The sampling has a 5% bias towards choosing the goal as a sample.



(a) 57 m, 2.98s, 1901 exp

(b) 75 m, 21.056s, 5401 exp

Fig. 60: Example of paths found by RRT in the second scenario: underwater terrain along with the length and time to find a solution and number of expansions.

4.4 Reparation Using Previous Explored Tree

Navigation in unknown or partially unknown environments may lead to the current plan being obstructed by an unknown object. If the vehicle is equipped with sensors able to detect such an event, the path needs to be updated. This can be done by either re-planning the path from the current state with an updated map based on the new detected obstacle or by repairing the current path. While planning a path to a goal many explored paths will not lead up to the solution. However, if a path has to be repaired due to a new obstacle which blocks the current path has been obsereved, one of the other previously examined paths might be useful. In the event that an obstacle is observed which blocks the current path, the planner repairs/re-plan the path by re-using previously examined paths. Before a repair starts, the tree is pruned from the branches which would lead to collision with the new obstacle(s) similar to work by Bekris and Kavraki in [148]. The flowchart for the reparation by pruning and re-using a previously explored tree can be seen in Fig. 61.

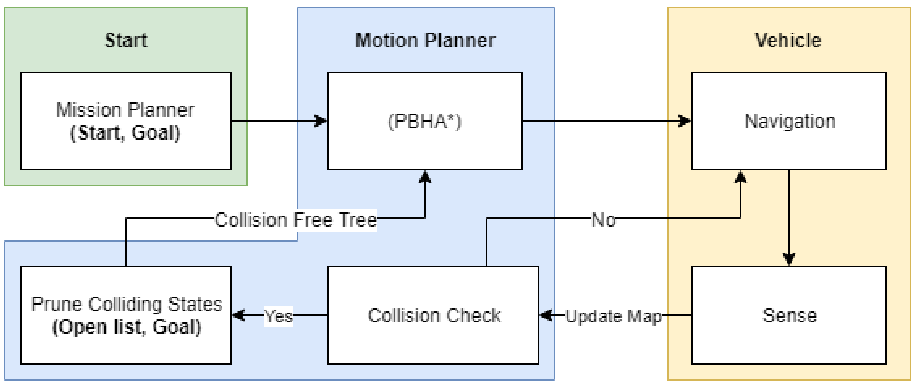
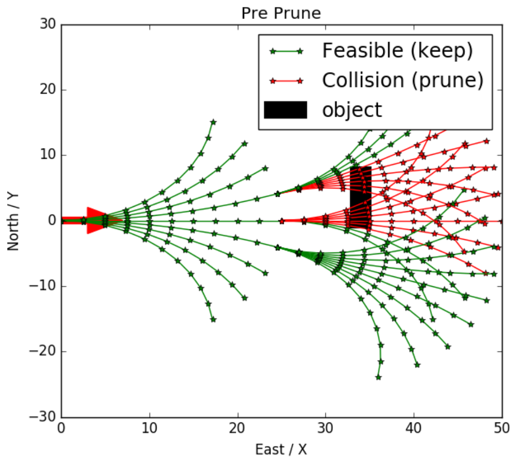


Fig. 61: The architecture for PBHA* which allows repairing of a path using previously explored states by pruning branches colliding with newly discovered obstacles.

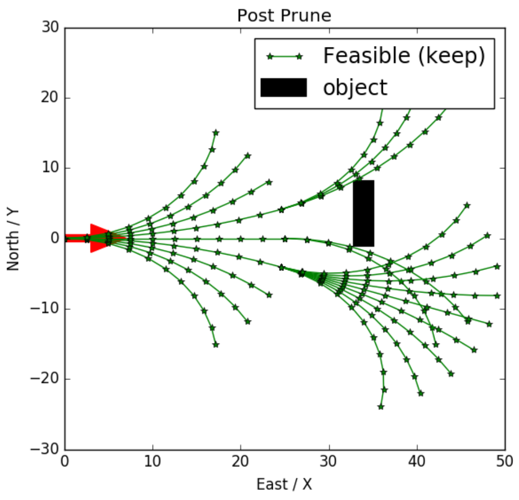
The pruning procedure needs to trim the tree from all states which are not usable anymore. These are both the states which will lead to collision in the future and all states that depend on them. The past states which have already been visited (or are presently being depended on) should also be removed from the tree. A visualisation of this can be seen in Fig. 62.

4.4.1 Reparation Simulation in 2D

To visualise the reparation using previously explored trees, a 2D scenario was designed. This scenario is based on a vehicle planning its path with PBHA* with pruning capabilities using the same pattern as shown in Fig. 62. The map for the scenario is completely unknown to the robot, as can be seen in Fig. 63a, where the vehicle first plans a straight path to the goal region. In the image, the grey obstacles are currently unobserved. When a parts of an obstacle are observed, an enlarged version of the observed object is added



(a) An object/obstacle has been observed. The tree is checked and non-feasible paths are pruned



(b) The pruned tree is used as input for the path planning algorithm

Fig. 62: The vehicle (red arrow) has detected an object which is blocking the path. In (a) it checks which paths are obstructed and prunes the tree of those branches. In (b) the remaining tree after being pruned is used as input for the planner.

to ensure a safety margin for the vehicle [100]). Each time the vehicle's path is found no longer be feasible, it uses the tree from the current state as the initial tree to plan a new path from. As can be seen in Fig. 63, the vehicle after some reparation manages to get to the goal region without any collisions. The vehicle in this simulated case is equipped with a sensor able to measure 250 units with a 60° field of view.

The re-planning using previous explored paths for the presented scenario reduces the number of expanded nodes compared to re-planning from an empty tree. For the described scenario the amounts of expansion to find a path can be seen in Fig. 64.

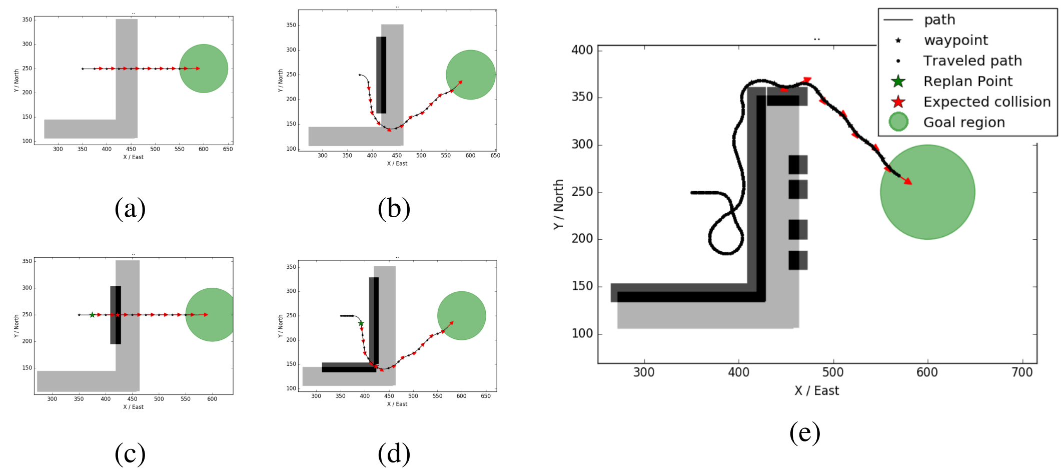


Fig. 63: Planning and repairing in an unknown environment. The grey area is an obstacle which has not yet been observed. When observed the vehicle adds that part with a static uncertainty bound around the observation. (a): initial planned path, while executed in (c) it notices a collision will occur and needs to react to create a new path. Some time later in (b) it has started to turn south and will later observe more of the obstacle in (d) and will start a path going north again until it in the end (e) it finishes a path around the obstacle. The simulated vehicle is equipped with a forward looking sonar with a 60° degree field of view.

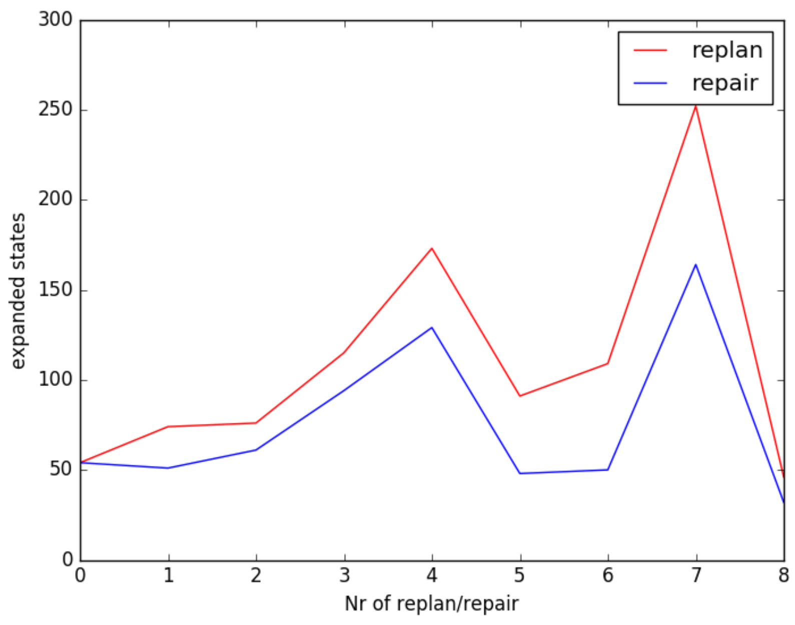


Fig. 64: Re-planning using the previous explored states reduces the number of expansions needed to find a solution. This graph is for the simulation shown in Fig. 63.

4.5 Summary

This chapter has presented a search based method to solve start-to-goal path planning under motion constraints.The potential motions are discritised into a pattern, which is a set of feasible trajectories for the vehicle. The chapter shows how exploration guided by weighted heuristic can be used to find a feasible path in real-time in 3D scenarios for non-holonmoic vehicles. It also shows how keeping previously explored paths during the execution of a path can help to reduce the time to re-plan/repair a path if a obstacle blocking the current path is observed.

A big advantage of the planner is that it is *resolution complete*, meaning that it will be able to find, and report if a solution exists. This is however based on the resolution of both the discrete grid as well as the resolution of the supplied motion pattern. A supplied set of potential motions reduces the complexity of the usage, as it can easily be changed and adapted to different types of vehicles as well as adding more possible trajectories gives a higher resolution. The presented planner shows a decreased time when compared to an RRT using the same type of motions for expanding a node in the tree. The property of being resolution complete

This planner have great benefits for planning in the local space of the vehicle. The size of the local space as well as the resolution of the grid where the planning is performed does however need to be based on vehicle type, range of sensing capabilities and potentially the environment. As a rule of thumb, for safe operations the vehicle should always have the possibility to come back to its current state without collision, in other words to move in a circle. For a small vehicle that can perform a circle with a small radius this is not as important as for example using the same algorithm on a tanker (which could have a minimum turning radius in the scale of kilometres). The same parameters needs to be taken into account for the resolution — a large ship does not need to know the environment in small details in the size of centimetres for planning a feasible path, while for a small vehicle knowing the map in great detail might help it navigate through small spaces.

The approach has, as it share many properties with A*, a drawback which is that it will always expand the unexpanded state with the lowest cost (path cost + heuristic cost). This makes it prone to get stuck in local minimas, until the whole area of the local minima is explored until it finds a path around it. A potential way to overcome this could be to add a wave-propagation from the goal to calculate the heuristic based on the environment instead of only the Euclidean distance [139, 140].

5 Vehicle Tracking and Following under Kinodynamic Constraints

The previous chapter described path planning under motion constraints in the traditional way of finding a path from X_{start} to X_{goal}. This chapter will instead focus on an objective driven path planner with the goal to reduce the distance between a leader vehicle (AUV) and a follower (CNA) from the follower's point of view.

This has previously been performed using different combinations of Proportional Integral Derivative (PID) controllers [94, 149, 150, 151]. However, such controllers assume that the following vehicle has a higher degree of motion capabilities in terms of controlling the vehicle's speed than the leaders. This is not always the case. Fig. 65 depicts a scenario where a surface vehicle equipped with engines forces it to drive continuously at a higher speed than the AUV can be seen. In such cases, classic control methods as PID controllers might not be the most efficient use or option, as the follower cannot be controlled enough to stay on top of the vehicle and will therefore overshoot and act unpredictable. Instead this chapter will suggest a path planning approach to following a leader. The aim for the planer is to take the motion constraints into consideration while finding a path which minimises the distance between the vehicles.

This path planner takes a discrete set of feasible trajectories into consideration to plan a path which, under the circumstances, minimises the distance between the two vehicles over time. It takes some elements from chapter 4 into consideration, such as a search tree based on exploration using a pattern.

The benefits of having a surface vehicle in proximity is that it can enable more reliable and higher speed communication, acoustic and in certain optimal conditions, optical. Section 5.4 will describe a scheduler for transmission times to improve the localisation on receiving AUVs. The scheduler is based on estimating the path of both the leader and follower over time to determine when the most beneficial Time of Launch (ToL).

5.1 Leader Position Estimation Based on Kinematic Model

To follow a submerged vehicle, acoustic communication is needed for the leader (AUV) to update the follower (ASV). Acoustic communication is however expensive and often

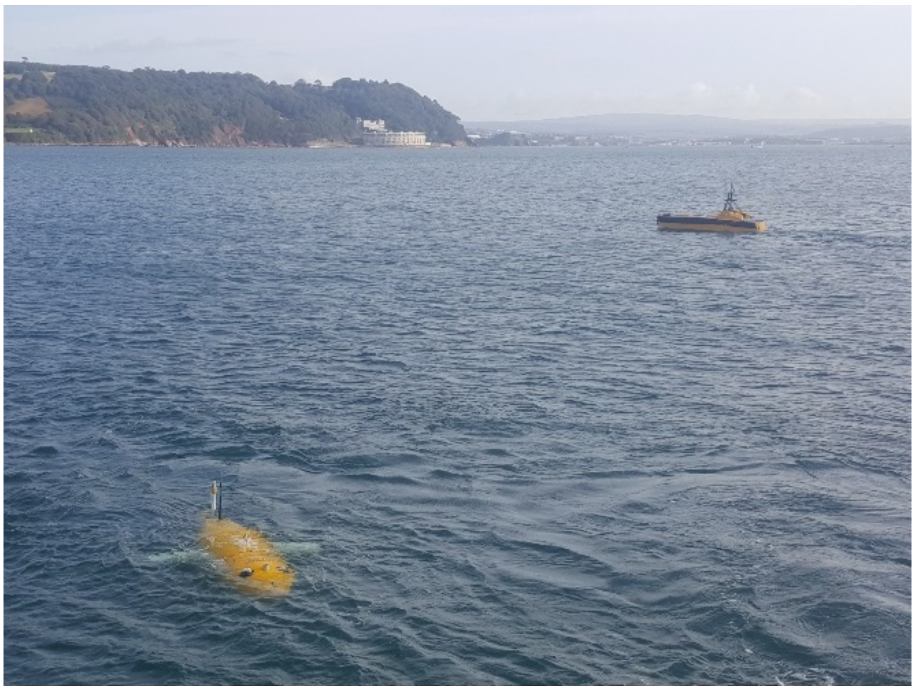


Fig. 65: Cooperative mission between C-Worker 5 (equipped with engines which forces it to drive continuously at a minimum speed) and Autosub Long Range (a slow moving vehicle at roughly 0.5 knots).

sparse[1]. Therefore, continuous and reliable updates from the leader is not always realistic. Instead the follower needs to be able to estimate the position of the leader vehicle over time in-between received updates.

The work presented here is opportunistic and event driven. That is, it does not assume any *a priori* information about the leader. Instead, it will wait or loiter until a leader has been observed, through receiving an acoustic message from a leader which includes its state estimate, and after that start to follow the vehicle. This has the benefit that no pre-mission coordination between vehicles is required and ensures that that the follower vehicle can work continuously in an area and wait for a leader to appear. When information from a leader vehicle is received (from any form of communication) the follower will update its target model which contains the leader's estimated position, the time of transmission, velocity, orientation and/or target waypoint. Using this data the follower can estimate the target's position over time. While the communicated data does not necessarily give a precise representation, especially long-term with multiple waypoints or if transmitted during turns (without including waypoints), it is typically the type of information that an AUV would transmit periodically[2].

To estimate the target (T) vehicle a simple kinematic model is used. The notations used are the same as in Fig. 34. If tracking is in 2D (49) is used. If a waypoint is provided this is used to calculate the yaw (ψ) value and if not, the vehicle's transmitted value is used. The same notations as in Fig. 34 are used, however V_{xy} is the speed in the plane

[1]As described in section 2.2

[2]This is based on SeeByte's autonomy framework Neptune [152]

and ψ is the estimated direction the vehicle is travelling at.

$$T(\Delta_t) = \begin{bmatrix} 1 & 0 & \Delta_t * V_{xy} & 0 \\ 0 & 1 & 0 & \Delta_t * V_{xy} \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ cos(T_\psi) \\ sin(T_\psi) \end{bmatrix} \tag{49}$$

If tracking is performed in 3D equation (50) is used.

$$T(\Delta_t) = \begin{bmatrix} 1 & 0 & 0 & \Delta_t * V_{xy} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta_t * V_{xy} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta_t * V_{xy} \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ cos(\psi) * cos(\theta) \\ sin(\psi) * cos(\theta) \\ sin(\theta) \end{bmatrix} \tag{50}$$

If the follower is aware of the target's current waypoint it can estimate when the leader will arrive at that position as in equation (51).

$$T_t^{WP} = \frac{||T_{WP} - T(0)||}{V} \tag{51}$$

If no new messages have arrived at time T_t^{WP} the leader will assume that the vehicle is static at the waypoint and use this as the position estimate until new data arrives. This is to safeguard the follower from continuing in a direction which would be opposite the leader's, causing the distance between them to grow faster and out of communication range earlier. It is therefore better to try to be close to the last estimated position of the leader until new data arrives.

5.2 Leader-follower Path Planner

The implementation of the leader-follower planner has some similarities to PBHA*, presented in chapter 4. Much like it, the presented path-planner for following a leader uses priority based expansion of a search tree. This is performed by expanding a motion pattern to the un-expanded state in the search tree with the lowest cost, see Fig. 67, iteratively until a termination condition is achieved. To this point, the algorithm is very similar to PBHA*, however it differs in multiple aspects. First, it uses a different type of cost function to prioritise nodes for expansion in the search tree. Second in this planner, the

C-Space is not used as a discrete grid as this would deny the possibility of a plan to re-visit the same cell which for a leader-follower scenario might be desirable. Third, the termination condition is based on finding a path for a specified duration instead of reaching a goal configuration or region.

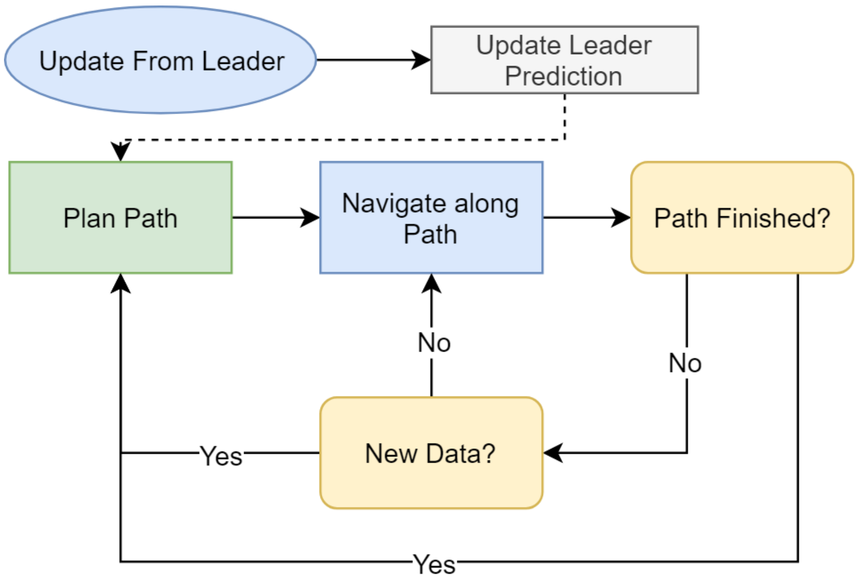


Fig. 66: Flowchart for kinodynamic leader-follower path planner. The planner updates the path based on new information received from the leader.

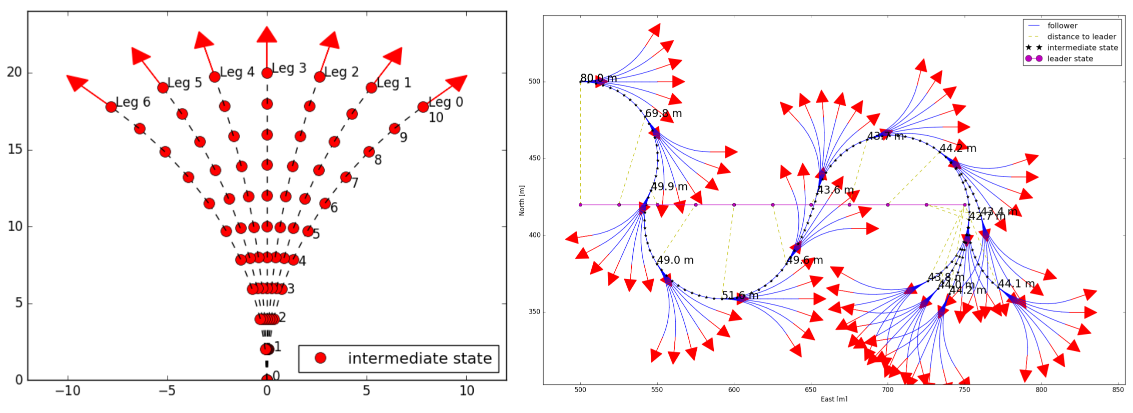
As can be seen in the algorithm's flowchart in Fig. 66, the planner is driven by updates from the leader. The algorithm to plan the path can be seen in Alg. 5.

Algorithm 5 Kinodynamic Path Follower

Input:
X_{start} : start configuration
T : Target (Leader)
$\mathcal{O}$: Obstacles, exclusion zones etc.
Δ_{goal} : Length (time) of solution
Φ : Motion pattern

```
1:  procedure KPF
2:      $openList$ = PriorityQueue()
3:      $openList.insert(X_{start})$
4:      while $openList$ do
5:          $q_{exp} = openList.pop()$
6:          if $q_{exp}.time \leq \Delta_{goal}$ then
7:              return $q_{exp}$
8:          for each $\Phi_i \in \Phi$ do
9:              $\hat{q}_{new}, Valid = expand(q_{exp}, \Phi_i, \mathcal{O})$ /* See Alg. 4 */
10:             if $accepted$ then
11:                 $openList.insert(f(\hat{q}_{new}), \hat{q}_{new})$
12:     return $\emptyset$
```

Prioritising which node to expand is based on the average between the follower and leader over time. This is performed by estimating where the leader would be at the same



(a) (b)

Fig. 67: A motion pattern as in (a) is applied iteratively to the state in the tree which has the lowest cost as in (52) until the termination condition is reached.

time as the follower would reach a new configuration. To obtain the average distance between the vehicles we summarise the distance between the leader and the follower along the path that leads to the current estimation to get a summed distance. The summed distance is then divided by the time it takes to execute the path. This can be seen in equation (52). In the equation $f(q)$ is the current evaluated state, n is the depth of the state and $\hat{T}_i$ is the estimated position of the target at the same time the leader is expected to reach state q_i ($\hat{T}_i = T(q_i^{time} - q_0^{time})$).

$$f(q) = \frac{\sum_{i=1}^{n}(||q_i - \hat{T}_i||)}{n} \tag{52}$$

The algorithm is continuously expanding until a state is opened which would take Δ_{goal} seconds to reach. As the algorithm always prioritises expansion of the lowest cost node (lowest average distance between leader and follower), a state which is opened and fulfils the termination condition is the best possible solution to be found under the current circumstances. An example of how Alg. 5 is applied with a pattern, as seen in Fig. 67a, can be seen in Fig. 67b. In this simulation the termination condition is fulfilled when a state is opened which has a depth of 10 in the search tree.

As there may be potential obstacles (including other vessels or too shallow water) or exclusion zones this should be taken into account while planning a path. As in PBHA* a collision check is performed as shown in Fig. 54 before a potential branch of the pattern is added to the search tree. An example of how the planner finds feasible and safe paths around an exclusion zone can be seen in Fig. 68.

5.3 Results

The path will be highly dependant on both the provided motion pattern of the follower (ASV) and the estimation of the leader (AUV). This has been tested with various configu-

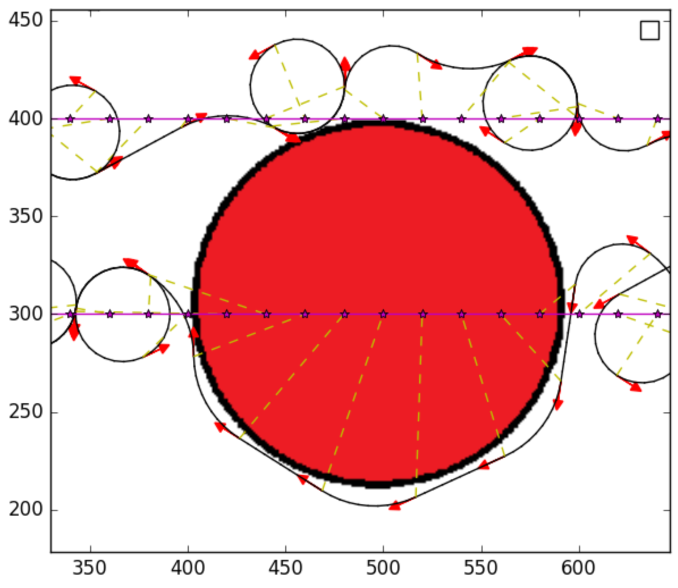


Fig. 68: The follower plans a path to avoid the exclusion zone (red circle) while aiming to minimise the distance between to the leader.

rations when it comes to the freedom speed and turning radii of the follower. An example when the follower has a very limited set of actions can be seen in Fig. 69.

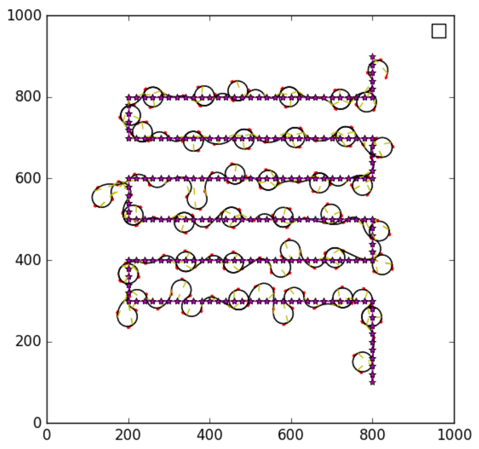


Fig. 69: The follower (ASV) vehicle's path while driving continuously at 5m/s while the leader (AUV) is has a constant speed of 2 m/s. The follower is able to turn with 25 or 50 metre radius and plans in steps of 10 seconds per motion. This results in a path that performs *sinusoidal* curves and encircles the leader. The follower estimates the leader's position over time based on the acoustic messages containing information about the leader's position, velocity, heading and goal position.

The same scenario as in Fig. 69 has been tested with various configurations of freedom for the follower. These are described in table 9. It comes to no surprise that the average distance between leader and follower decreases with the higher number of branches in the motion pattern and the amount of discrete speeds that the follower is able to use.

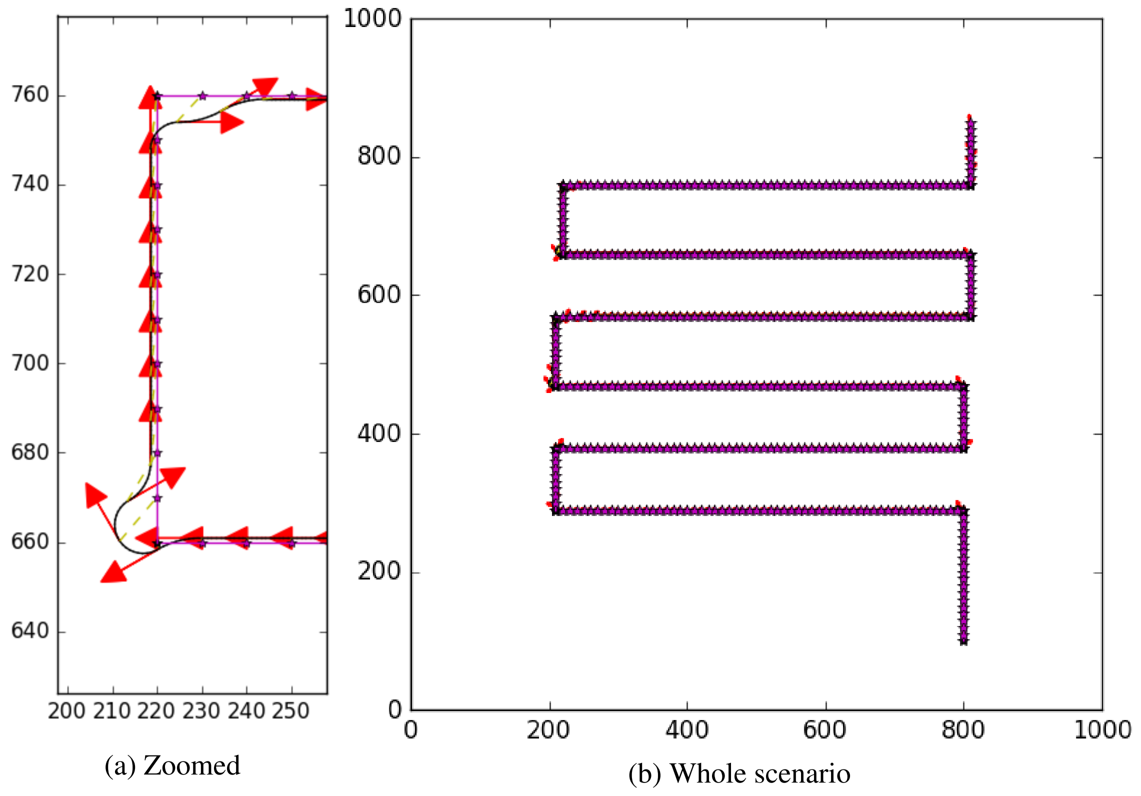
Kinodynamic Path Follower - Results ASV following AUV			
Turning radius (m)	number of branches per pattern	possible speeds for follower (m/s)	Avg. dist between leader and follower (m)
$25, 33, 50, 100$	9	0, 1, 2, 3, 4, 5	5.5
$25, 50$	5	2, 2.5, 5	7.20
$25, 50$	5	0, 5	17
$25, 50$	5	2.5, 5	19.1
$25, 50$	5	2.5	24.2
$25, 50$	5	2	32.2
$25, 50$	5	5	33.6

Tab. 9: Path following comparison - different motion constraints. The leader performs the lawnmower shown in Fig. 69 with a constant speed of 2m/s.

In table 9 we can observe a row where the follower has the same speed as the leader (2m/s) and the resulting path becomes similar to what would be achieved by a Proportional Integral (PI) controller such as in [94, 153] as the follower vehicle will aim to drive in the same direction as the leader until the leader communicates that it has changed direction and waypoint. The follower then needs to do a turn which results in falling behind the leader. This is because if they drive at the same speed it can never catch up with the current estimates but instead, when the leader changes direction once more, the follower can cut a corner to decrease the distance by a bit. A subset of the path from this scenario can be seen in Fig. 70.

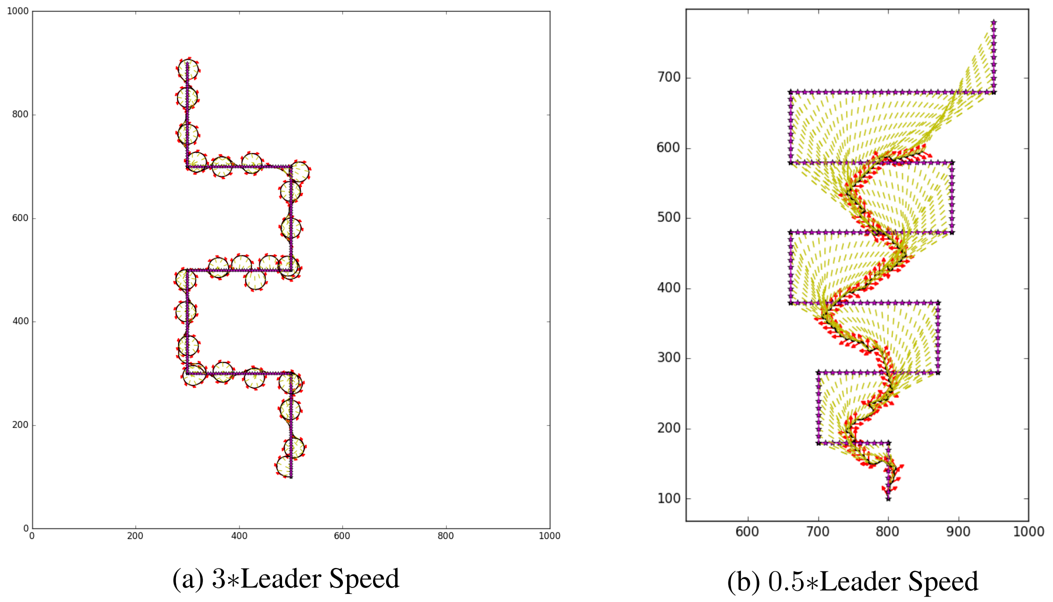
5.3.1 Results 3D

The same algorithm can easily be applied to any dimension by changing the supplied input to be in the required C-Space. An example has been performed in $\mathbb{R}^3$ where a follower vehicle able to move with a similar pattern as in Fig. 53 is used to follow a surveying AUV. The resulting path, as seen in Fig. 72a and 72b, becomes much like a helix around the target. Similar results are achieved when the target is only going downwards, in which case the follower will perform a spiral motion downwards as seen in Fig. 72c. In a real life maritime environment, these types of leader follower scenarios probably do not have much use but nonetheless, it shows that the same algorithm is adaptable to multiple dimensions.



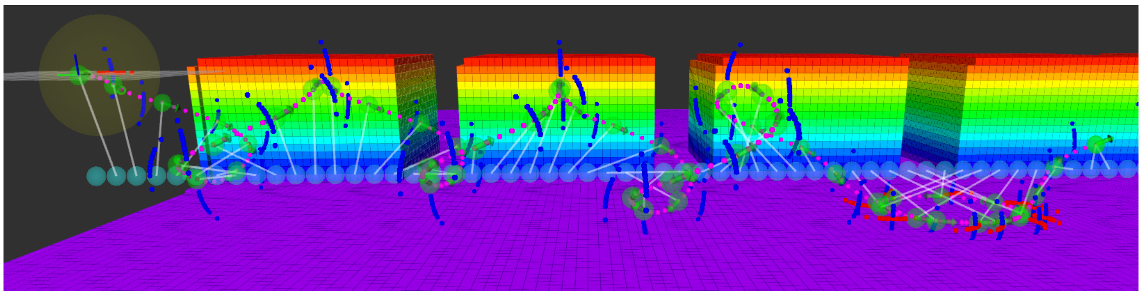
(a) Zoomed (b) Whole scenario

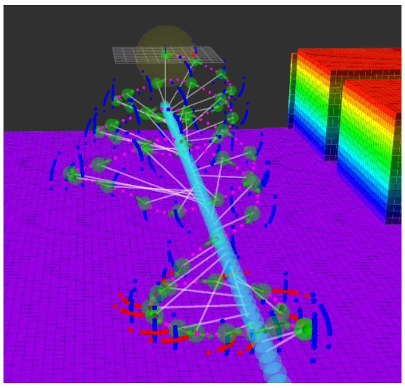
Fig. 70: When the follower operates at the same speed as the leader it tends to fall behind after the leader changes direction but can take the next direction change as an opportunity to catch up.



(a) $3{*}$Leader Speed (b) $0.5{*}$Leader Speed

Fig. 71: Example of different paths created based on the follower's speed compared to the leader.


(a) Helix - Side view


(b) Helix - Front view

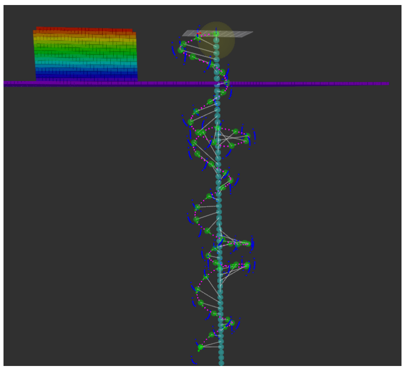

(c) Spiral - Side view

Fig. 72: The leader-follower path planner in $\mathbb{R}^3$. The follower creates a helix like pattern to reduce distance over time to the leader.

5.4 Selection of ToL to Improve Acoustic Localisation

The leader-follower path planner presented in this chapter would be suitable to use as CNA to support a submerged AUV. This is as it ensures the creation of feasible paths while maintaining the goal to minimise the distance to the follower over time. This should improve the quality of the acoustic signal[3] and hence making the acoustic communication more reliable. Acoustic communication can be used for localisation as described in chapter 2 where it can be observed how the found position is effected by different geometric relationships between the transmitter and receiver. The most beneficial geometric relationship between the transmitter (CNA/follower) and receiver (AUV/leader) is dependent on the localisation method. In this section two methods will be considered, EKF and NLS. A brief summary of how these methods can be improved by the geometric relation ship can be seen below or in chapter 2.

- For **NLS** the optimal configuration is when the transmitters are evenly spread (angle wise) around the receiver [28, 72, 154].
- For **EKF** the closer a transmission is located to the semi-major axis of uncertainty from the covariance matrix, the greater the reduction of uncertainty will be [29, 57].

In order to consider a planning approach as presented in this chapter it will create a path to follow a target and hence now the estimated geometrical relation between the vehicles over time. This is as it knows where the follower will be and can estimate the position of the leader based on the kinematic models in equation (49) or (50). Assuming that a Time-Division Multiple Access (TDMA) protocol (see Fig. 73) is used to control at which time-slot a vehicle can transmit data from, an informed decision could be made about when to transmit a message used for localisation at the time in which the geometrical relationship should be more useful.

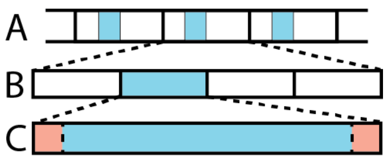


Fig. 73: TDMA is used to divide time among units to reduce message collision on a shared channel such as acoustics in water. **A**: Time is divided into frames. **B**: A frame consists of time slots. **C**: A time slot is the time a platform has the possibility to transmit and has an optional guard time in the beginning and end to avoid collisions from other time slots.

This section presents a framework to select the times for transmission, ToL, in a way which over time should reduce the localisation error on receiving platforms more than the conventional way where messages are transmitted periodically.

[3]As stated in section 2.2 the signal strength decreases with the distance travelled

5.4.1 Selection of Transmission Time

The outline of the presented planner for transmission time can be seen in Fig. 74. It is based on updating estimations of the AUV based on what information it transmits. The information is the same as what the leader transmits during the leader-follower scenario, with an added 2×2 covariance matrix if the localisation method used is EKF. The CNA is also assumed to know its predicted plan ahead in time. Based on both the AUV and the CNA's estimated positions over time, this information is being sectioned into sets based on when the CNA is allowed to transmit within the TDMA.

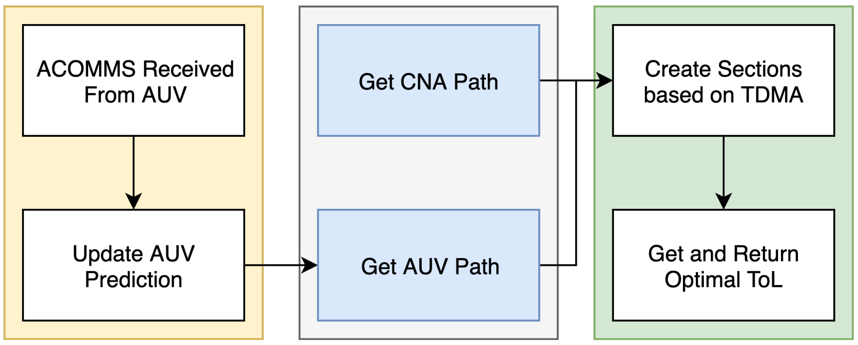


Fig. 74: The selection of ToL planner is a layer used to find transmission time for acoustic localisation message based on positional estimates of the vehicles and the TDMA.

The framework uses the different sections containing the geometrical relationships between the vehicles to find a set of transmission times which, based on the vehicle estimates, should reduce the error the most. The selection method is based on finding a combination of 1 estimate from each set which minimises a cost function. This cost function differs based on which localisation method the AUV uses.

5.4.1.1 Extended Kalman Filter Cost Function

The motivation behind the cost function for EKF is explained in section 2.6. The cost function for each geometrical relationship is the residual angle from the vector between the vehicles and the semi-major axis of uncertainty of the ellipse derived from the transmitted covariance matrix.

5.4.1.2 Nonlinear Least Squares cost function

The cost function for NLS is based on what has been shown earlier — the more evenly (angle-wise) the transmission positions are spread around the receiver — the better the solution should be. The NLS problem needs more measurements than the dimension of the solution is in to find a unique position. This cost function is based on the minimum amount of measurements required, which for a solution in $\mathbb{R}^2$ is 3. Based on this, a cost function which the lowest cost while the 3 different geometrical relationship is equally

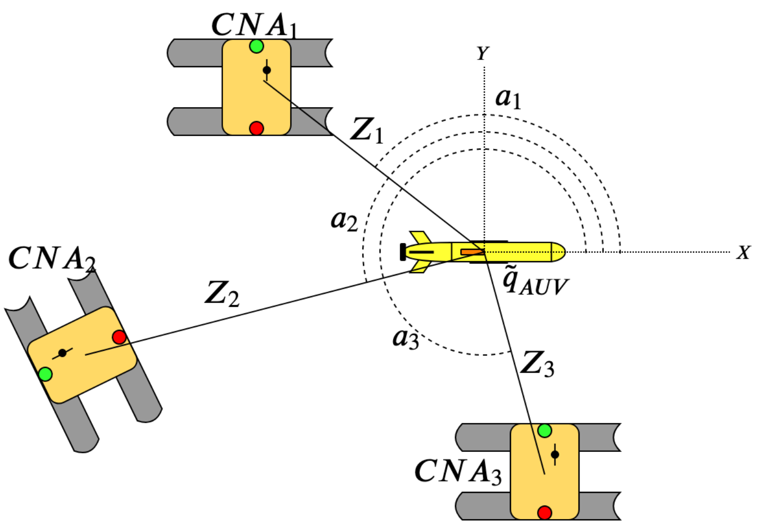


Fig. 75: The polar angles between the AUV an the CNA over time. These angles can be used in equation (55) to estimate how well the localisation algorithm on the AUV would perform. The 3 different CNAs does not necessarily be different vehicle but can also be the same, but at different times.

spread out, and the worst when the relationships is such that they all or on the same axis. The function to minimise can be seen in equation (53).

$$a, b, c = argmin(f(\alpha, \beta, \gamma)\forall_{\alpha \in A, \beta \in B, \gamma \in C}) \tag{53}$$

Equation (53) returns the best configuration from each of 3 sets. This is based on evaluating the different combinations (one from each of 3 sets) as in equation (55). Equation (55) uses equation 54 which takes the 3 estimated polar angles (a_1, a_2, a_3) between the AUV and the CNA as parameters.

$$\begin{aligned} f(\alpha, \beta, \gamma) = &|cost(\alpha, \beta, \gamma) - cost(\beta, \alpha, \gamma)| \\ &+|cost(\beta, \alpha, \gamma) - cost(\gamma, \alpha, \beta)| \\ &+|cost(\gamma, \alpha, \beta) - cost(\alpha, \beta, \gamma)| \end{aligned} \tag{54}$$

$$cost(a_1, a_2, a_3) = CW(a_1, a_2, a_3) + CCW(a_1, a_2, a_3) \tag{55}$$

In (55) *CW* and *CCW* return the minimum residual angle from a_1 in the clock-wise respective counter clock-wise direction to a_2 and a_3. An example of how these angles can look like can be seen in Fig. 75.

An example of the cost in different configurations of angles can be seen in Table 10 and in Fig. 76. A visual example where three sets of angles are evaluated to find the solution which minimise (53) can be seen in Fig 77b.

α	β	γ	cost(α, β, γ)
0	$2\pi/3$	$-2\pi/3$	0
0	$\pi/2$	$-\pi/2$	π
0	$\pi/6$	$-\pi/2$	$7\pi/3$
0	$\pi/6$	$-\pi/6$	3π
0	0	0	6π

Tab. 10: Example of the cost (equation (54)) between the three angles α, β and γ

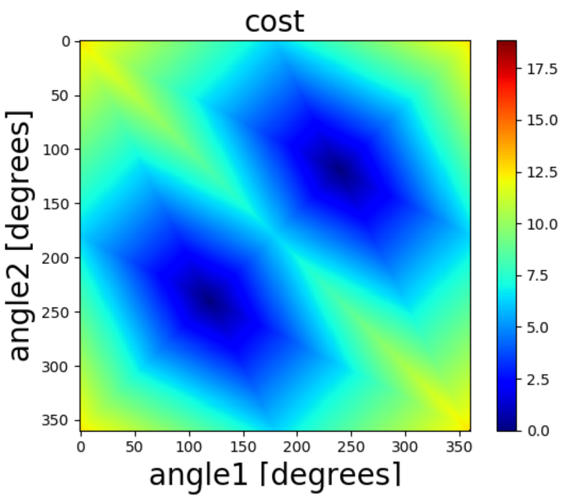
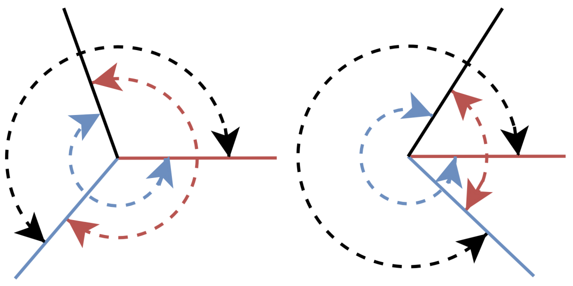
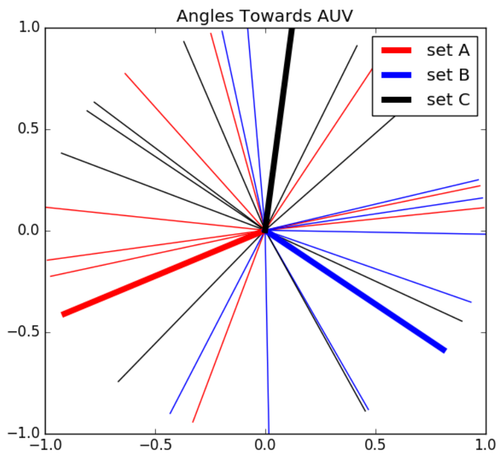


Fig. 76: The cost associated with tree angles where one angle is referenced as 0 degrees in equation (14). It can be seen that the lowest cost is achieved when the 3 angles are equally distributed as also is shown in table 10.



(a) The function used to find optimal transmission positions for NLS compares the clockwise and counter-clockwise residual to the other evaluated angles which the estimated positions of the vehicles would create. (Left) evenly distributed angles create good conditions to solve the NLS problem, while (right) is a worse setup to solve the problem as can be seen in Fig. 19.



(b) 3 Sets of the angles from geometrical configurations between an AUV and an ASV. The thicker lines show the 3 configurations which create the most even distribution between landmarks, and hence should produce the most accurate result.

Fig. 77: The geometrical relationship that produces the lowest error for the trilateration problem is when the angle (relative direction) of the received messages' positional origin are as evenly distributed around the receiver as possible, as shown in Fig. 19.

5.4.2 Paths Effect on Variety of Geometrical Relationship

The presented method of planning a path in a leader-follower scenario during different motion constraints, such as different speeds, can lead to a large variation of the geometrical relationship over time. Other methods where different types of P,I and/or D controllers [94, 149, 150, 151] have been used to keep the follower vehicle *on top* of the leader might reduce the variance in the geometrical relationship. An example of how the variation over time can be for distance between the vehicles and the relative angle can be seen in Fig. 78. In this figure, two cases are presented, the first (top) is when the follower has a constant speed which is twice the leaders. In the second case (the bottom) both vehicles operate at the same speed (an example of such a scenario can be seen in Fig. 70).

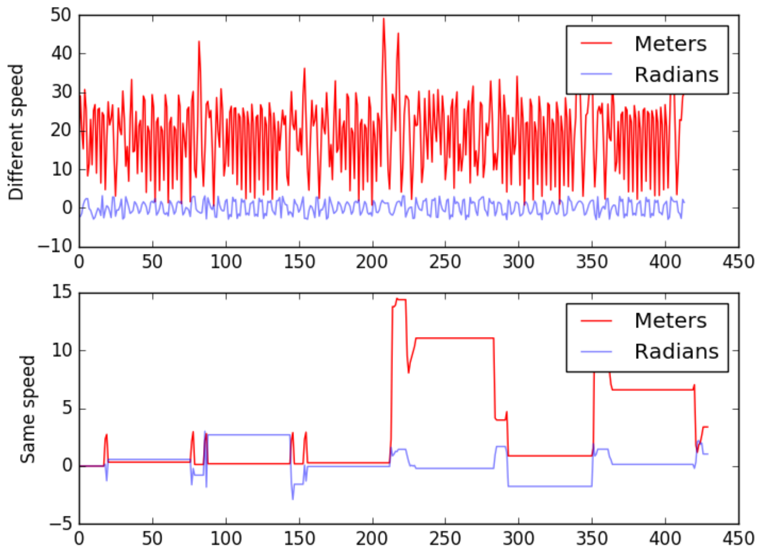


Fig. 78: Different speed of the leader and follower vehicle increases the variance in the geometrical relationship. This can be used to reduce the error in acoustic localisation.

As can be seen in the second case, where the vehicles operate at the same speed, there is a low variance in the geometrical relationship over time compared to when they operate at different speeds and the follower needs to vary much its path to stay close to the leader (see Fig. 71 for an example of such a path).

In the case where the two vehicles operate at the same speed and the geometrical relationship over a time-slot can be expected to be rather static there might not be much improvement by selecting the transmission time. In the first case there is much variance over time, so the variety of geometrical relationships to select from is large, and as such the improvement in the receiver's localisation should also be able to be improved more.

5.4.3 Results Using AUV Sirius Dataset

To test the selection of ToL algorithm the dataset collected by AUV Sirius was used. It has been compared to the conventional way of transmitting periodically (static time within a time-slot). For the comparison a TDMA with a considered slot length of 20 seconds and two frames are examined (one each for CNA and AUV Sirius). The AUV updates the transmitter every 4:th frame (160 seconds period), and hence the CNA needs to use proposed ToL selection method to find 4 transmission times. For the comparison a moving transmitters (ASVs) is considered where it has different objevices

The ASV and distance measurements are simulated. The noise in the simulated distance measurements has a Gaussian distribution with a σ of 2.5 and 5 metres. The static transmission times within the TDMA slot are 0, 10 and 19 seconds. The different scenarios evaluated (which can be seen in Fig. 79) are: an ASV performing a lawnmower survey, circling the area and the leader-follower path planner presented in this chapter.

The average error on the AUV is reduced in all simulated cases as seen in Table 11 and 12 using out adaptive selection of ToL compared to the static ToLs. In all cases the transmitting platforms have no prior knowledge about the AUV. They are adapting based on the incoming acoustic messages containing the AUV's estimated position, heading and velocity. When the localisation method is EKF, a 2×2 covariance matrix is included. In the table it can also be seen how the cost function performs by selecting the estimated best time of transmission compared to how it performs using static times. It can be seen that for the adaptive approach, the cost function as well as the navigational error on the AUV is the lowest, no path smoothing was performed on the AUV. It can be seen that a lower cost function in general leads to a lower positional error on the AUV. Which supports the proposed cost functions usability. In the scenarios where the variance in the polar angle between the vehicles is small (circle and lawnmower) the positional error grows larger than the leader-follower scenario where the polar angle between the vehicles has a larger variance over time.

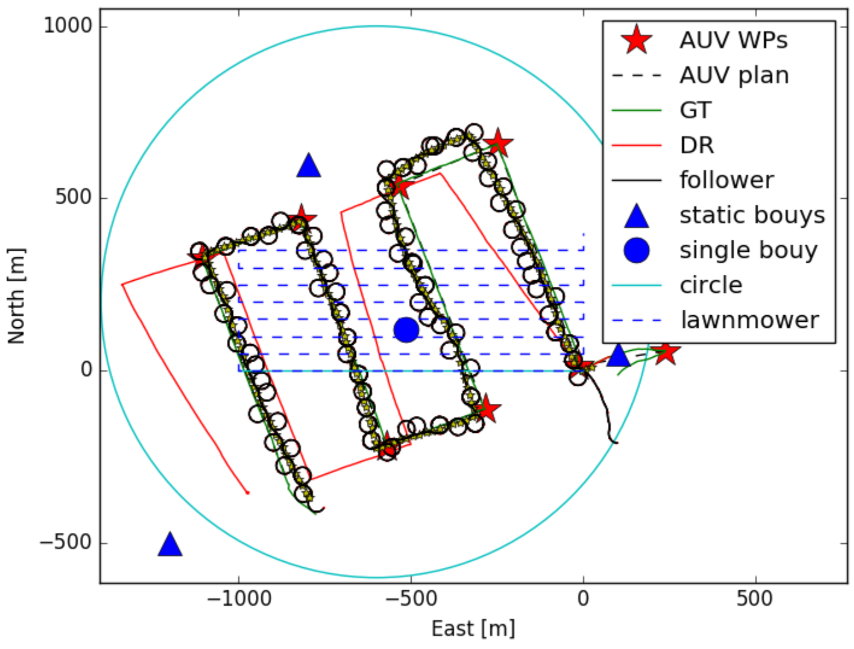


Fig. 79: Example of different paths and configurations in the evaluated scenario. Follower is the path planner presented for the leader-follower scenario.

Extended Kalman Filter : Leader-follower				
ToL	Adaptive ToL	first (t=0)	mid (t=10)	last (t=19)
Distance measurement error [m] = $\mathcal{N}(0.0,\ 2.5)m$				
AUV Sirius Error (Avg [m])	5.2	8.1	7.9	8.3
Cost function (Avg)	0.09	0.79	0.77	0.82
CovMat Max Axis 1 std (Avg)	2.66	2.94	2.90	2.98
Distance measurement error [m] = $\mathcal{N}(0.0,\ 5.0)$				
AUV Sirius Error (Avg [m])	8.5	10.0	9.6	9.7
Cost function (Avg)	0.09	0.81	0.78	0.74
CovMat Max Axis 1 std (Avg)	2.65	2.96	3.01	2.95
Monte-Carlo Gradient Descent (NLS) : Leader-follower				
ToL	Adaptive ToL	first (t=0)	mid (t=10)	last (t=19)
AUV Sirius Error (Avg [m])	5.6	8.6	7.9	7.7
Cost function (Avg)	1.98	5.50	5.13	5.46
Distance measurement error [m] = $\mathcal{N}(0.0,\ 5.0)$				
AUV Sirius Error (Avg [m])	8.8	9.8	10.4	11.9
Cost function (Avg)	2.38	5.73	5.22	5.26
Extended Kalman Filter : Circle				
Distance measurement error [m] = $\mathcal{N}(0.0,\ 2.5)$				
AUV Sirius Error (Avg [m])	10.8	12.0	11.5	12.1
Cost function (Avg)	1.39	1.43	1.42	1.41
CovMat Max Axis 1 std (Avg)	5.39	5.39	5.39	5.39
Distance measurement error [m] = $\mathcal{N}(0.0,\ 5.0)$				
AUV Sirius Error (Avg [m])	11.7	12.8	13.0	12.7
Cost function (Avg)	1.39	1.43	1.41	1.40
CovMat Max Axis 1 std (Avg)	5.37	5.38	5.38	5.38
Monte-Carlo Gradient Descent (NLS) : Circle				
Distance measurement error [m] = $\mathcal{N}(0.0,\ 2.5)$				
AUV Sirius Error (Avg [m])	14.0	19.6	14.2	14.6
Cost function (Avg)	11.5	11.7	11.5	11.5
Distance measurement error [m] = $\mathcal{N}(0.0,\ 5.0)$				
AUV Sirius Error (Avg [m])	23.7	29.23	32.8	23.7
Cost function (Avg)	11.6	11.7	11.7	11.6

Tab. 11: The correlation between the cost function and the error on the AUV. The lower the cost, the closer a transmission from a position on a more beneficial polar form the AUV, as such a lower cost should result in a lower error which can be seen. The TDMA window considered is 20 seconds long, hence the first, mid and last transmission is sent at t=[0, 10, 19] and the adaptive approach presented selects a time between 0 and 19 for transmission. The error from DR is 195.4 metres.

Extended Kalman Filter : Lawn mower				
ToL	Adaptive ToL	first (t=0)	mid (t=10)	last (t=19)
Distance measurement error [m] = $\mathcal{N}(0.0,\ 2.5)m$				
AUV Sirius Error (Avg [m])	5.9	7.6	6.9	7.0
Cost function (Avg)	1.32	1.38	1.37	1.35
CovMat Max Axis 1 std (Avg)	4.86	4.87	4.87	4.87
Distance measurement error [m] = $\mathcal{N}(0.0,\ 5.0)$				
AUV Sirius Error (Avg [m])	8.8	9.9	10.3	8.9
Cost function (Avg)	1.32	1.38	1.36	1.35
CovMat Max Axis 1 std (Avg)	4.8	4.8	4.8	4.8
Monte-Carlo Gradient Descent (NLS) : lawn mower				
Distance measurement error [m] = $\mathcal{N}(0.0,\ 2.5)$				
AUV Sirius Error (Avg [m])	7.3	23.6	18.3	23.6
Cost function (Avg)	11.3	11.47	11.38	11.32
Distance measurement error [m] = $\mathcal{N}(0.0,\ 5.0)$				
AUV Sirius Error (Avg [m])	29.0	44.4	41.0	30.6
Cost function (Avg)	11.21	11.37	11.32	11.25

Tab. 12: Continuation of Table 11.

5.5 Summary

This chapter have presented two algorithm related to improving cooperative navigation between marine robots. The first one is a path planner for a leader-follower scenario. This planner takes motion constraints (including speed) into consideration to plan a path for the follower to reduce the average distance between the leader and the follower over time. This has great benefits in scenarios where the vehicles have different motion capabilities. An example is when a leader is a slow moving vehicle and the follower has a minimum speed which exceeds the leader's maximum speed. In such cases, classic control theory as PID controllers fail to supply a reliable solution. The proposed method is a priority based expansion of a search tree. Expansion of a state in the tree is based on both the motion constraints and the velocity of the follower. The proposed algorithm shows reliable real-time planning capabilities for cases including when the follower operates at; lower, same and faster speed than the leader. The algorithm is used for a single-leader, single-follower scenarios. Having more leaders would not be impossible as the cost function could be the sum of the cost for all leaders. This would however make the follower follow the estimated middle point between all leaders. Another approach could be to add a growing reward cost when a leader is not the one being followed, so that either a travelling salesman problem could be solved to select which leader to focus on or to always follow the leader with the highest reward function.

The second part of the chapter is a ToL selection layer which runs on top of a path planner. By estimating both the transmitter's and receiver's position over time, strategic transmission times (and hence positions) for acoustic localisation messages can be chosen. By selecting a ToL in which the geometric relationship between the vehicles is estimated to be the most beneficial a reduction in positional error on the receiving platform can be achieved compared to periodic transmission strategies. This proposed adaptive method is compared towards static transmission times of acoustic localisation messages within a time window and shows to at worst perform equal to static transmission times. In most cases the adaptive approach has reduced the error on the AUV by up to 35%. As this approach does not need to alter the path or behaviour of the transmitting vehicle in anyway more than control when to transmit a message it would be beneficial to use as a scheduler on acoustic transponders without demanding much computational power. This does however at the moment not take the depth of the AUV into consideration and the results presented is based on shallow operations.

6 Spatio-temporal Path Planning to Improve Range-only Localisation

"I may not have gone where I intended to go,
but I think I have ended up where I needed to be."

- Douglas Adams

This thesis has to this point presented various path planning algorithms and acoustic localisation methods, along with ways to improve this based on estimation of the geometric relationship between transmitter's and receiver's position. The previous chapter presented a path planner for multi-vehicle scenarios and a planner to improve acoustic localisation by selection of transmission time. In that chapter they are considered to be independent of each other. This will have its benefits as the CNA can perform its own survey and use the ToL planner to improve localisation on other AUVs. This chapter will present an approach on how these can be intertwined from a path planning perspective. It will present a path planner for a single CNA in which the objective is to act purely to support submerged AUVs with acoustic localisation messages.

6.1 Planning to Improve Localisation

A spatio-temporal planner plans both in space and time. Planning in both these dimensions are beneficial for the purpose of reducing the error and uncertainty related to acoustic localisation methods. Previous planners for CNAs to position themselves at beneficial geometric relationships to AUVs have been used considering periodical transmission, and hence planning only in space. Planners such as the one shown in Fig. 32 (see section 2.9) presented by Bahr *et al.* [29] samples the whole reachable region of the CNA to select the best position[1] as the next waypoint for transmission. Munafo *et al.* instead use a constant speed vehicle and include external forces (e.g. currents) to sample points on the perimeter of a circle (radius and offset depending on vehicle speed and external forces) to calculate the determinant of the FIM (equation (20)) and select the most informative position as

[1]Based on EKF, the transmission position should be as close as possible to the semi-major axis of uncertainty.

the next waypoint [71]. Both these plan for a single step ahead in time, while the presented planner looks over a longer horizon to find a set of waypoints along with times for transmission which are aimed to reduce the uncertainty over time. Looking at multiple steps ahead in time can prevent the CNA from e.g. choosing a next transmission position which also will put the CNA in a position where it will fall behind the AUV and limiting its options for the next transmission [29]. Planning for multiple steps can therefore lead to reducing the uncertainty to a greater extent — as a combination of waypoints which should reduce the uncertainty even greater can be found.

Instead of sampling points and selecting the most beneficial one, other approaches have used parameterised trajectories. This includes zigzag[22, 155], circles [22, 57] and diamonds/squares[155]. However, these do not consider the estimated movement of the AUVs. The movement pattern is more or less static and pre-defined instead of adapting to the estimation of the AUVs. Dynamic Programming [57] and Markov Decision Process [89] have been implemented to plan the path of a CNA to aim to transmit messages at the semi-major axis of uncertainty of the AUV, This shows reduced uncertainty compared to more static CNA trajectories.

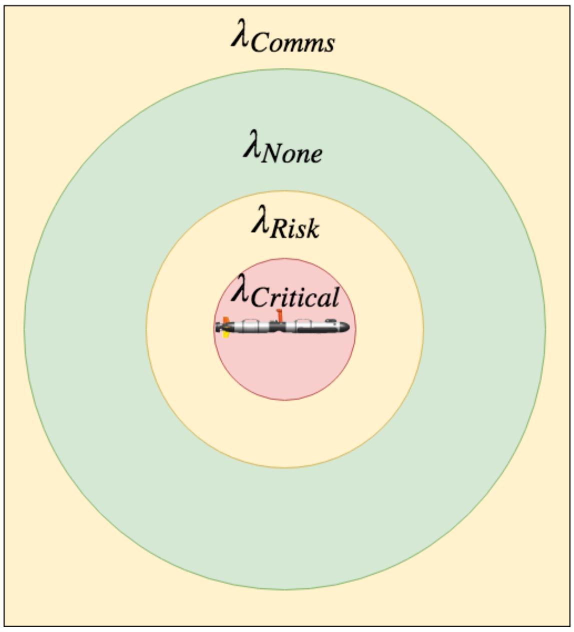


Fig. 80: Different risk zones can add various penalties while planning the path of a CNA. This is to reduce risks of potential collisions if the AUV surfaces and to keep the vehicles within communication range.

Safety is a concern for cooperative missions like these; vehicles should avoid actions which increase the risk for collision. As such, the CNA is encouraged to operate within a minimum and a maximum distance from the AUV to avoid collision and to stay within communication range by heavily penalising paths outside the acceptable boundaries. This is considered by Hudson *et al.* who apply penalties depending on inter vehicle distance while using look-ahead (planning a few steps ahead) and pre-decided trajectories to plan a path for the CNA to support multiple AUVs by adding the best of the trajectories to the search tree in each iteration [156]. The method can be used both with pre-planned missions on the AUVs or as a response to received acoustic messages. Similar penalties are also used in [89] and [57]. An example of such regions, which are applied in the presented algorithm in this chapter can be seen in Fig. 80. The penalty function can be constructed in various ways, for example it can be based on the depth of the AUVs such that the penalty is the lowest in a region which is at a distance such that the slant angle towards the AUV is large by using work by Chen *et al.* [91, 70].

6.2 Path Planner to Improve Acoustic Localisation

The presented planner in this chapter is a hybrid between priority based expansions (such as A*) and sampling-based exploration (as RRT). The flowchart of the algorithm can be seen in Fig. 81 and the pseudo-code in Alg. 6.

The planner applies the same expansion process as other priority based search trees presented in this thesis. It expands a tree until a termination condition has been reached. The order the tree expands its nodes in is based on a cost assigned to each node. The termination condition is when the planner expands a node with a specified depth ω (in the tree). This represents finding a path consisting of ω waypoints and is returned as the plan for the CNA to follow until new information is received which might update the plan. When a node q_{exp} is expanded, its offsprings' cost are added with the cost of q_{exp}. The cost function is based on the residual angle between the CNA and AUV to the semi-major axis of uncertainty for the AUV. Hence the cost can never be less than 0. As a cost of a node can never be lower than its parent, always expanding the lowest cost state ensures that when the termination condition is reached (node's depth$=\omega$) it cannot be improved under the current parameters of the algorithm. The cost function is based on that the receiving AUVs using EKF as a method of localisation. It has been described earlier in section 2.6 and 5.4 that the geometrical relationship between transmitter and receiver which reduces the uncertainty by the most is when a transmission occurs along the semi-major axis of uncertainty [76, 29, 57]. This axis is derived from the covariance matrix on the AUV. As such the CNA needs to receive the 2×2 covariance matrix from the AUVs along with data to be able to estimate their positions over time by using the same kinematic equations presented in section 5.1.

There are two events which require the algorithm to calculate a new list of waypoints. This is either by receiving an update from an AUV or if the last waypoint of the current list of waypoints is reached. To run the algorithm the following are required:

- X_{start} - current state of the CNA and its knowledge about the AUVs ($X^{Targets}$).
- ω - number of waypoints to plan for (depth of a state in the tree).
- N - number of random-sampled states to add to a node under expansion.

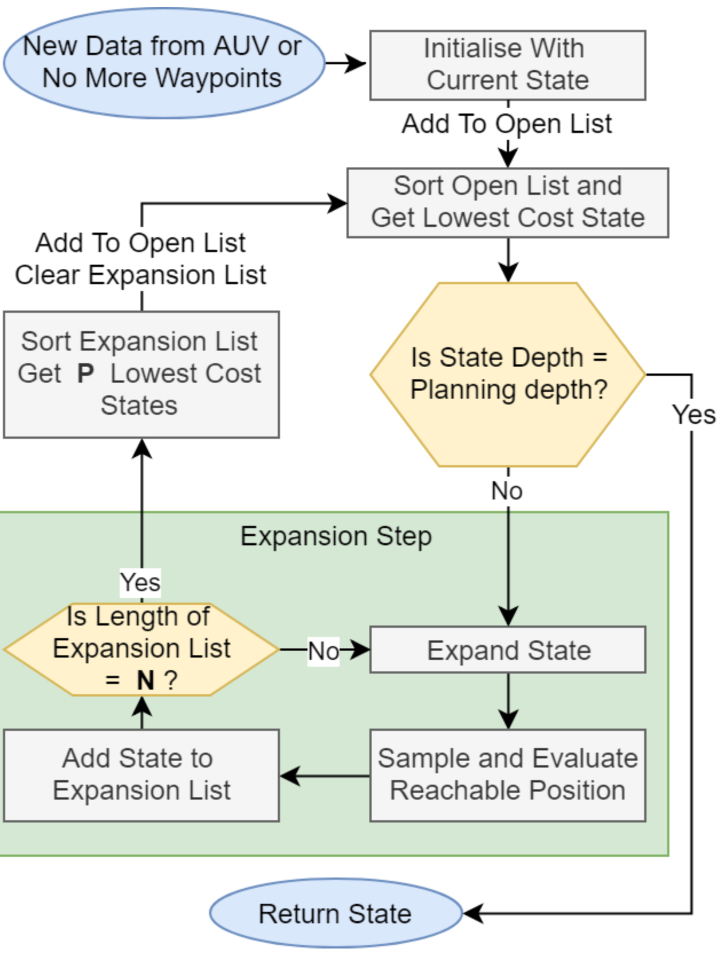

New Data from AUV or No More Waypoints
Initialise With Current State
Add To Open List
Sort Open List and Get Lowest Cost State
Add To Open List Clear Expansion List
Sort Expansion List Get P Lowest Cost States
Is State Depth = Planning depth?
Yes
No
Expansion Step
Yes
Is Length of Expansion List = N ?
No
Expand State
Add State to Expansion List
Sample and Evaluate Reachable Position
Return State

Fig. 81: The presented path planner is a combination of priority- and sampling-based methods to find a path consisting of a defined number of waypoints and optimal time for a CNA to transmit localisation messages to AUVs.

Algorithm 6 Sample-Based Transmission Waypoint Planner

Input:
X_{start} : start configuration
ω : Number of waypoints to find
N : Number of expansion
P : Number of expansions after pruning

```
1: procedure PLANNER
2:     $Open = priority\_queue([X_{start}])$
3:     while $Open! = \emptyset$ do
4:         $X_{open} = Open.pop(0)$
5:         if $X_{open}^{depth} == \omega$ then
6:             return $X_{open}$
7:         $\hat{X} = Expansion(X_{open}, N, P)$
8:         for each $x \in \hat{X}$ do
9:             $Open.add(x)$
```

- P - prune the list of N newly expanded states to only keep P.

As stated, the algorithm is a spatio-temporal planner. The *spatio* part relates to space and the position of a planned waypoint. The *temporal* part of the planner comes from selecting the ToL within the TDMA in which the estimated relationship from the CNA's waypoint is as close as possible to the semi-major axis of uncertainty on the AUVs. This selection process can be seen in Fig. 82 and a description of TDMA in Fig. 73.

6.2.1 Expansion of States

The expansion procedure of a state is sampling-based. The algorithm for expansion is described in Alg. 7. It is based on sampling a set Ψ consisting of N random positions in a region around X_{exp}. The region in which the sampling takes place is dependent on the CNA's maximum speed, the current time and the latest time the CNA can transmit a localisation message from within its next TDMA slot. From the current time and the latest time possible to transmit we get a duration. By multiplying this duration with the maximum speed we get a distance, which corresponds to the radius of the circle around X_{exp} in which the sampling takes place. This is to ensure that a sampled position is reachable before it needs to transmit an acoustic message from it. A visualisation of the expansion procedure can be seen in Fig. 82.

Algorithm 7 Expansion Step

```
Input:
$X_{exp}$ : Configuration to expand
$N$ : Number of expansions
$P$ : Number of expansions after pruning
 1: procedure EXPANSION
 2:     $Y = priority\_queue()$
 3:     $\Psi = X_{exp}.sample(N)$
 4:     for each $\Omega \in \Psi$ do
 5:         $ev = priority\_queue()$
 6:         $T = TDMA(X_{exp}, \Omega)$
 7:         for each $t \in T$ do
 8:             $\sigma = \sum_{\tau}^{X_{exp}^{Targets}} \zeta(\Omega, \tau_t) + \delta(\Omega^{pos}, \tau_t^{pos})$
 9:             $ev.add((X_{exp}, \Omega_t^{cost=\sigma}))$
10:         $Y.add(ev[0])$
11:     return $Y[0 : P]$
```

Below the functions and equations used in Alg. 7 are described,

- $TDMA(X_{exp}, \psi)$: This function returns a set (T) of discrete (1 second interval such that $T = [t_0, t_1, .., t_{n-1}, t_n]$) times based on the earliest time (t_0) the CNA could transmit from X_{New}^{Pos}. t_0 and t_n is described in (56) and (57). $TDMA_{n+1}^{0}$ is the earliest time in the next TDMA slot that the CNA is allowed to transmit from. A visual representation of this can be seen in Fig. 82.

$$t_0 = max(X^{ToL}_{expanding} + \frac{Distance}{CNA_{v\_max}}, TDMA^0_{n+1}) \tag{56}$$

$$t_n = TDMA^0_{n+1} + TDMA^{SlotTime} \tag{57}$$

- $\zeta(\Omega, \tau_t)$ is the residual angle Θ (58) which is in the range $[0, \frac{\pi}{2}]$. θ is the dot product between the vector of the semi-major axis of uncertainty of the ellipse which represents the estimated covariance matrix and the vector between position Ω and the estimated position of the target at time t, ($\overrightarrow{\Omega\ \tau^{pos}_t}$). The estimated position is calculated based on the same kinematic equations as presented in section 5.1.

$$\Theta = min(\theta, ||\pi - \theta||) \tag{58}$$

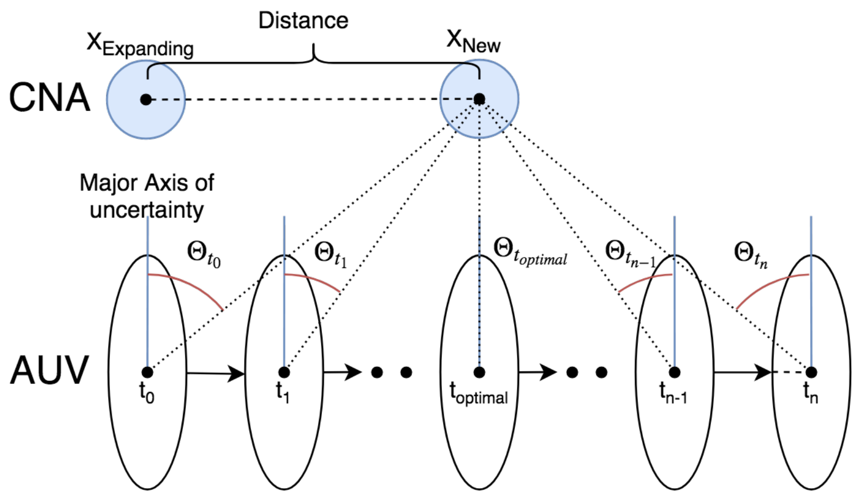
- The sampling function $X.sample(N)$ returns a set of N new reachable states from the position of state X. The random sampled points are within the distance $t_n * CNA_{v\_max}$

- The penalty function $\delta(\Omega^{pos}, \tau^{pos}_t)$ adds a penalty to a state if its waypoint is too close or too far from the AUV under evaluation. This is to reduce the risk of collisions between vehicles and aims to stay within communication range. The penalty is based on 4 different zones: *Critical*, *Risk*, safe (*None*) and if outside of communication range (*Comms*). This can be seen in algorithm 8 and a visualisation of the different regions can be seen in Fig. 80.

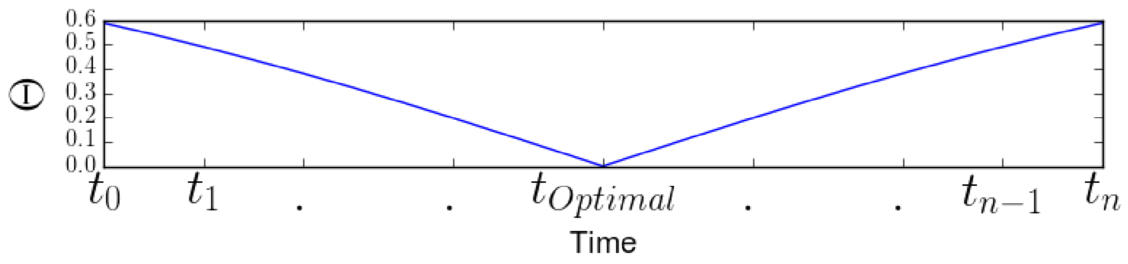
Algorithm 8 Inter vehicle distance based penalty

```
Input:
Ω : Position of CNA waypoint
τ : Estimated position of AUV
 1: procedure δ
 2:     $\Delta = ||\Omega - \tau||$
 3:     if $\Delta < \lambda_{Critical}$ then
 4:         return $\Lambda_{Critical}$
 5:     else if $\Delta < \lambda_{Risk}$ then
 6:         return $\Lambda_{Risk}$
 7:     else if $\Delta > \lambda_{Comms}$ then
 8:         return $\Lambda_{Comms}$
 9:     else
10:         return $\Lambda_{None}$
```

After the N new states have been added to the list Y of sampled expansions, the list is pruned to only keep the P lowest cost new states. This is performed as to reduce the search space.



(a) A sampled position (X_{New}) is evaluated to find the optimal ToL which is the t that results in the lowest Θ_t. Θ_t is the residual angle between the AUV's major axis of uncertainty and the vector between the estimated positions of the vehicles ((58)) in the time window where the CNA is allowed to transmit.



(b) The optimal time to transmit from $X_{New}^{Position}$ is the time which has smallest Θ.

Fig. 82: The expansion is performed by randomly sampling a new position and then evaluating when within the TDMA slot the optimal ToL for a ranging message would be.

6.3 Computational Efficiency vs Optimallity

The choice of using elements from sampling-based path planning is to reduce the search space enough to run the algorithm in real-time. While it is possible to expand a complete search tree to find the optimal solution, which is estimated to reduce the uncertainty the most, all reachable lattice points with all possible (whole seconds) ToL have to be considered in the search space. One iteration with a maximum reachable distance r metres creates a grid of all reachable lattice points as seen in equation (59) [157]. This is visualised in Fig. 83.

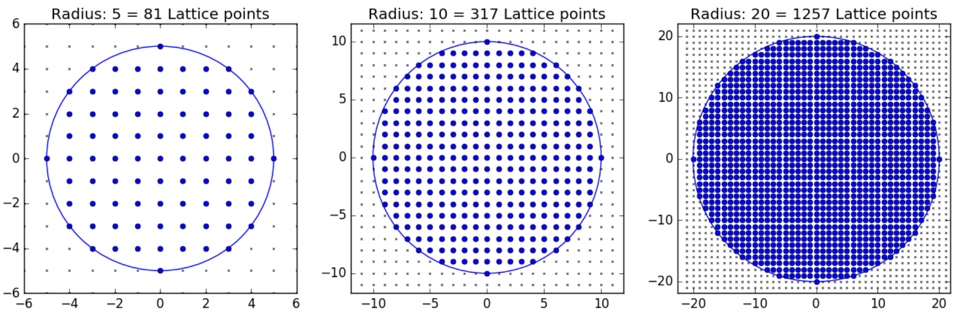
$$N(r) = 1 + 4 * r + 4 \sum_{i=1}^{r} (\sqrt{r^2 - i^2}) \tag{59}$$

By extending (59) to include all possible ToL at respective lattice points it results in all possible states for each iteration to achieve the optimal position and transmission time.

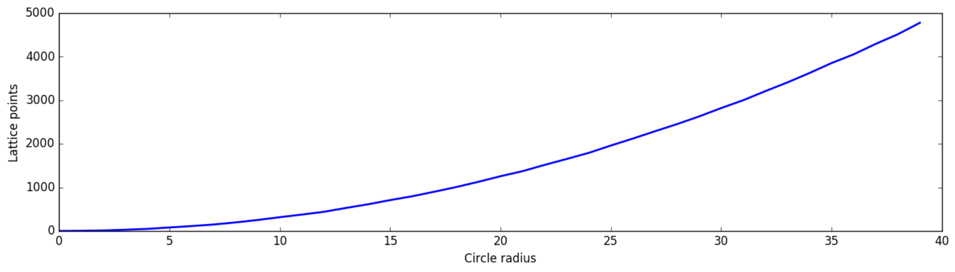
Each iteration expands the search tree by $O(T)$, where T is the time until latest possible ToL, nodes as seen in (59). To plan an optimal path of M waypoints the number of expanded nodes would be $O(T)^M$. This is in the case where the maximum speed is considered to be 1 m/s.

$$O(T) = \sum_{t=1}^{T} (T - t) * N(T - t) \tag{60}$$

$O(T)$ grows exponentially, while the proposed algorithm explores between N and $N * T$ and expands the search tree with the P most promising states each iteration. As such to evaluate all possible paths to find the optimal solution would be of many orders magnitude larger than the proposed approach. This reduction of the search space enables real-time computation, and as seen in Fig. 86a and 86b, the larger P and N, the better solution, at the cost of longer execution time.



(a) Lattice points for different radiuses at circles



(b) The number of lattice points grow exponentially with the radius of the circle

Fig. 83: The number of lattice points represent the possible integer valued positions within a circle needed to be evaluated to obtain the best transmission position (without considering time).

6.4 Results

The presented spatio-temporal path planner has been implemented and evaluated in a simulated environment using simulated navigational data as well as a navigational data set from AUV Sirius[2]. It has been compared to static trajectories using the traditional approach of periodical transmissions of localisation messages. It shows a decreased uncertainty and error towards these in all scenarios. As the algorithm can be used with various parameters an evaluation of these have also been compared. For the presented figures, the parameters used can be seen in table 13. All simulations were performed using Python 2.7 on an i7-7820 (2.9GHz) with 16 GB RAM. It is worth mentioning as python is an interpreted language, the algorithm would achieve a substantial speed-up using a compiled version of the code (e.g. *C++*), enabling it to run with a higher number of samples and keeping more states after the pruning to be able to achieve *better* results in the same execution time or the same results in less time.

Parameter	single	multi	Sirius
TDMA slot time	20	20	20
AUV speed [m/s]	1.5	1.0	0-0.652
P	3	5	5
N	100	50	25
ω	5	5	4

Parameters applied to all presented cases

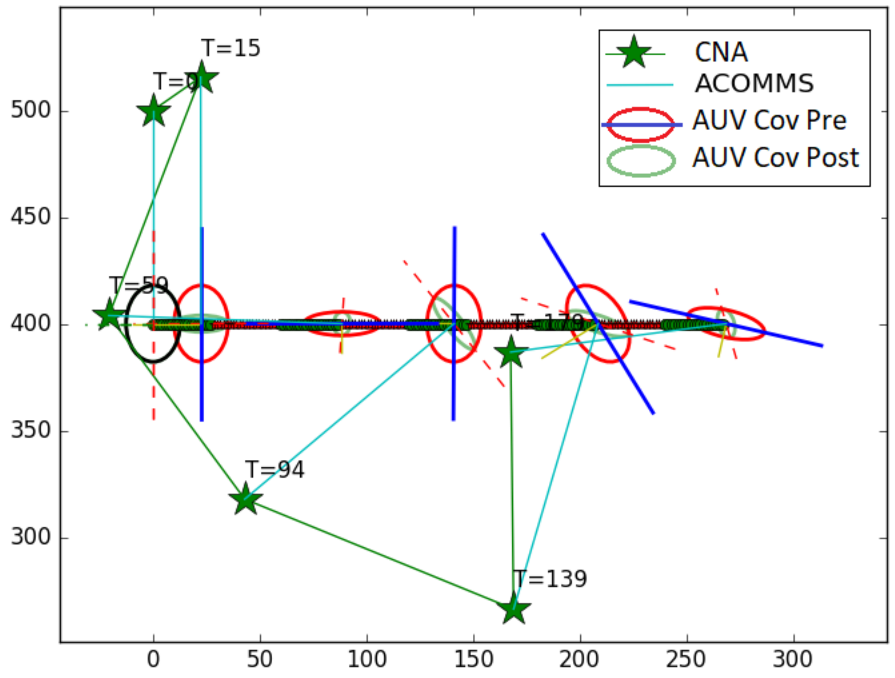
$\Lambda_{Critical,Risk,Comms,None}$	1.0, 0.5, 0.5, 0
$\lambda_{Critical,Risk,Comms}$	50, 100, 250
CNA max speed	3.0 m/s
TDMA slots	2

Tab. 13: Planning parameters used in simulations.

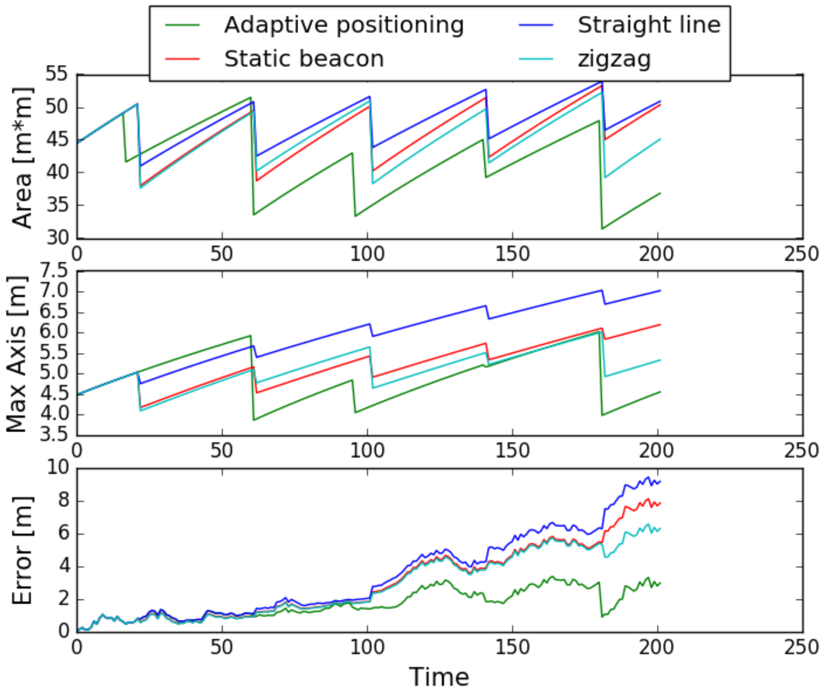
6.4.1 Single Simulated AUV

To display the path planned by the algorithm when a single AUV is operating can be seen in Fig. 84a. Part of a simulated survey leg performed by the AUV can be seen in the figure along with the path planned by the CNA. The path positions the CNA at a risk free waypoint for transmission. The path planned by the presented approach for the CNA shows an improved localisation when compared to other methods which can be seen in Fig. 84b. In the comparison the other methods transmits periodically. It shows improvement in the size of the ellipse representing the uncertainty, the size of the semi-major axis of uncertainty as well as the positional error.

[2]See section 2.8.3



(a) The (adaptive) path planned for a CNA supporting 1 AUV.

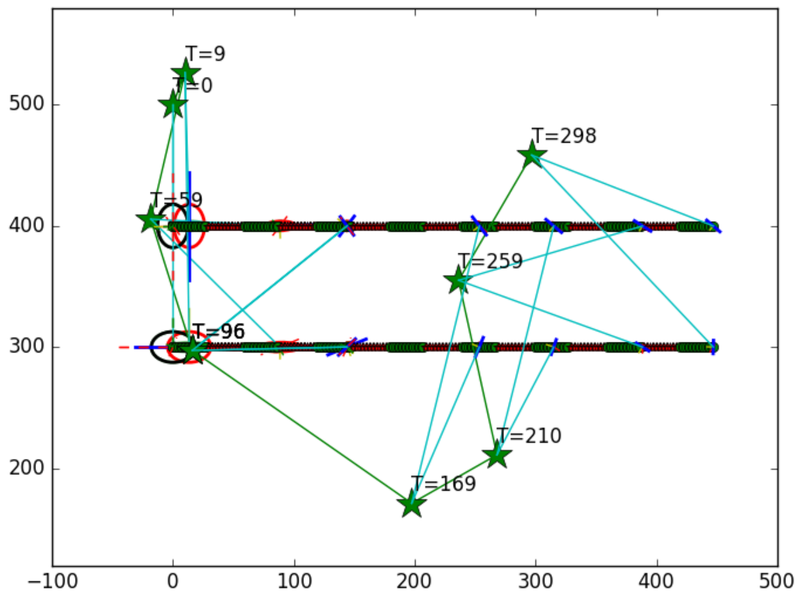


(b) Navigation error (bottom) alongside error estimates (area of covariance matrix and maximum axis of covariance matrix) on an AUV using range-only EKF.

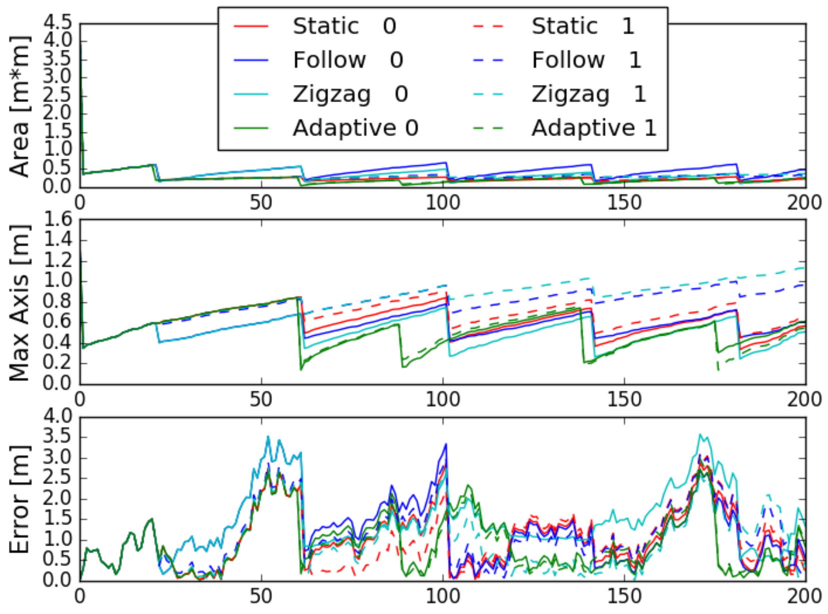
Fig. 84: The path of the CNA using the spatio-temporal path planner to find a path which aims to transmit along the semi-major axis of uncertainty for a single simulated AUV performing a survey. It show reduction in all compared metrics to other approaches. The parameters of the scenario can be seen in table 13. The stars are the waypoints for the CNA and the ellipses are the estimated covariance, the blue line is the semi-major axis of uncertainty and the light blue line shows the communication to each vehicle at that position. The black ellipse represents the initial uncertainty on the AUV.

6.4.2 Multiple Simulated AUVs

As the path planner is able to handle multiple AUVs, a scenario where the AUVs perform a survey was tested. The CNA's path and the AUVs can be seen in Fig. 85a. Just as for the single AUV a comparison towards the same methods was compared and the uncertainty and error from this can be seen in Fig. 85b. The proposed approach achieves a lower overall error and uncertainty compared to the other approaches. In multi-vehicle scenarios, the vectors representing the major axis of uncertainty might not intersect within the sampling region. Hence the cost function is the sum of the residual angle and penalty function for all vehicles. Instead of aiming to minimise the error on one AUV the planner will strive to minimise the sum of uncertainty on all participating vehicles.



(a) The (adaptive) path on a CNA supporting two AUVs.



(b) The uncertainty and error on vehicle 0 and 1.

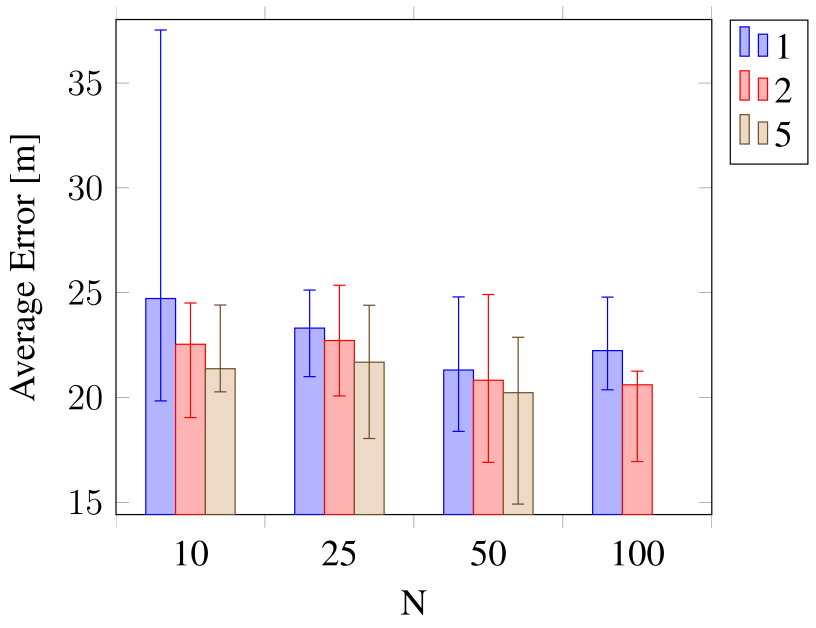
Fig. 85: The path of the CNA using the spatio-temporal path planner to find a path which aims to transmit along the semi-major axis of uncertainty for multiple simulated AUVs. It shows reduction in all compared metrics to other approaches. The parameters of the scenario can be seen in table 13. The stars are the waypoints for the CNA and the ellipses are the estimated covariance, the blue line is the semi-major axis of uncertainty and the light blue line shows the communication to each vehicle at that position. The black ellipse represents the initial uncertainty on the AUVs.

6.4.3 Dataset - Sirius AUV

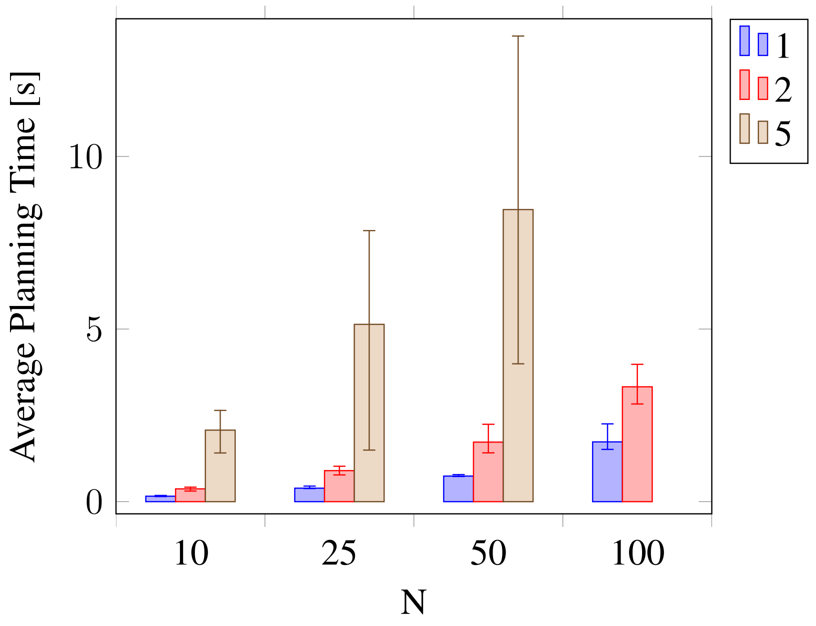
A dataset from Sirius AUV performing surveys outside Tasmania have been used to validate the effect of transmission from the planned waypoints. The data used for the EKF on the AUV integrates orientation and velocity from a DVL along with the simulated ranging measurements from the CNA. The AUV's ground truth is considered to be the localisation obtained from visual SLAM by Mahon *et al.* [16] combined with USBL. The paths of both vehicles can be seen in Fig. 87a and the resulting error in localisation on the AUV is compared to DR, a static beacon, a CNA following the AUV (with same speed and heading) and a CNA moving in a zigzag pattern which can be seen in Fig. 87b. The AUV transmits updates to the CNA with its estimated position, heading, velocity and 2-by-2 covariance matrix periodically every 160 seconds. The mean error for the proposed adaptive method compared to others can be seen in Table 14 and in Fig. 86a, where the average error of the proposed methods is reduced for all simulations compared to the other methods.

Method	Average Error
DR	183.0
Follow	168.6
Static	60.3
Zigzag	36.5
Adaptive	14.9-37.5

Tab. 14: Comparison of the mean error in meters on an AUV between proposed adaptive path planner and other approaches based on a dataset from AUV Sirius. The proposed adaptive approach reduces the error significantly by planning a path that aims to transmit messages from a position which is estimated to be close to the major axis of uncertainty on the AUVs. For more detailed display of the result of the proposed approach see Fig. 86a.

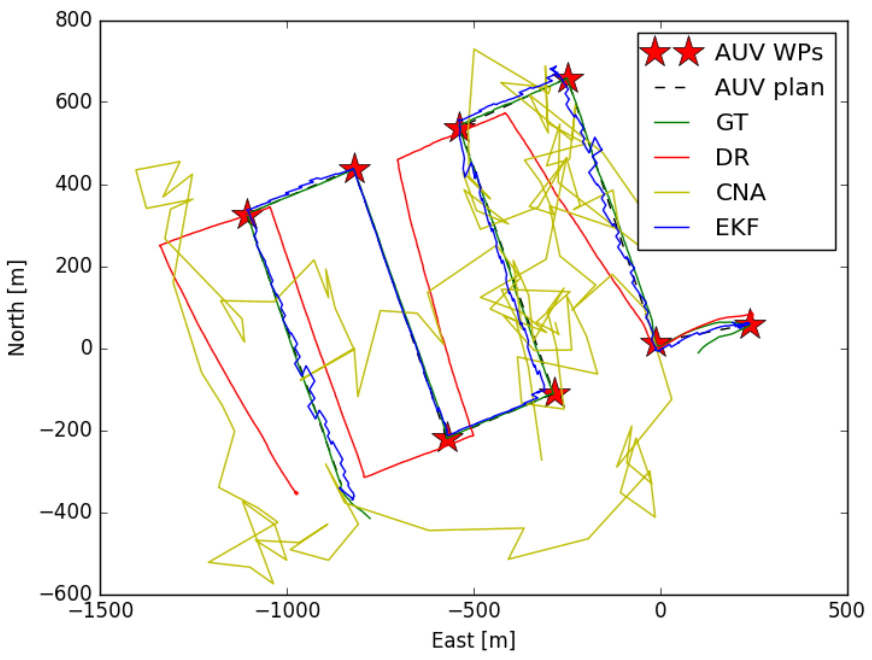


(a) The average error on AUV Sirius.

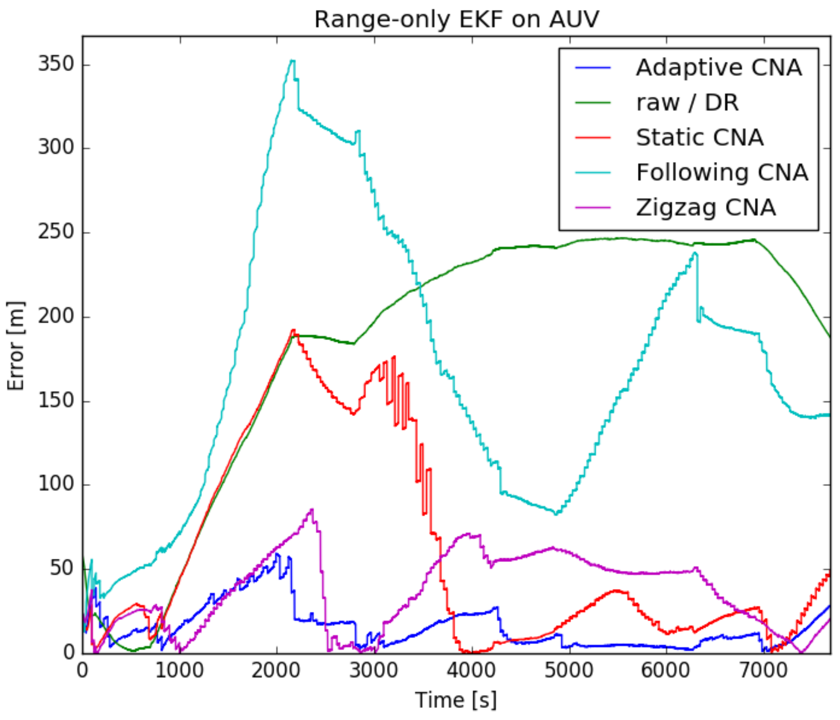


(b) The planning time depends on the number of expansions and samples.

Fig. 86: The average error and execution time to plan next 4 waypoints (∼160 seconds). Each set of parameters are simulated 10 times. The simulations use a navigational dataset collected on AUV Sirius combined with the proposed adaptive planning approach on a CNA. The different bars are the number of nodes saved from an exploration (The P value in the algorithms and flowchart), N is the number of samples during exploration. The average error on proposed approach is lower than the approaches it is compared to, as can be seen in Table 14.



(a) The adaptive path planning approach running with navigation data collected by AUV Sirius outside Tasmania, Australia.



(b) The resulting error on AUV Sirius with a range-only EKF, the presented (adaptive) approach is compared to other approaches and DR.

Fig. 87: A dataset from AUV Sirius combined with a simulated CNA.

6.5 Summary

This chapter have presented a novel approach to the problem of path planning for the scenario where a CNA supports an arbitrary number of AUVs with distance-measurements used for localisation. The planner is a hybrid between priority-based expansion of a search tree with sampling-based exploration. The approach assumes no prior knowledge about other vehicles. Instead it responds and adapts its path based on incoming acoustic messages from the AUVs. This makes it a dynamic and versatile approach, as a CNA running this planner can be deployed in the operational area of the AUVs without needing to be configured or informed about the AUVs' objectives and instead will start to support them when an acoustic message is received. The approach shows a great improvement towards other mobile CNA's trajectories. It plans a path a few steps ahead instead of choosing the next optimal position which could lead to reduced usage in the long run as the vehicle might fall behind the AUVs. To improve safety and reduce the risk of collision, and loss of acoustic range, it adds penalties paths which are either in risk of collision or outside of communication range. We show how using a limited number of samples instead of the whole search space reduces the time spent for planning enough to achieve real-time operations.

7 Conclusions

This thesis presents path planning methods for cooperative robotic scenarios in the maritime environment. It shows how to plan a path in real-time for a vehicle to reach a goal configuration and how to minimise the distance between two vehicles, both while taking the motion constraints of the vehicle into consideration. It also presents how to reduce acoustic localisation error by planning paths for a acoustic beacon to aid submerged AUVs. The thesis can be seen as a combination between acoustic localisation methods and path planning. This conclusion will summarise the different chapters and bring attention to the novel work it has lead up to both for publications and the thesis as well as ideas on how to extend the presented work.

7.1 Summary of Thesis

A review of acoustic localisation techniques can be found in chapter 2, where the focus is on range based localisation. This is as range-only localisation can, if using One-Way-Travel-Time, support an arbitrary amount of AUVs with a single message. This is followed by a chapter on path planning, where the two major fields are the focus. The first one is grid-based algorithms and the second one is sampling-based, two very different approaches which both have their benefits and drawbacks which will be taken advantage of in this thesis. Section 2.9 presents the concept of Communication and Navigation Aid, where a vehicle is dedicated to support other vehicles with acoustic messages to improve localisation as well as acting as a communication relay to be able to extend the communication for submerged vehicles to have a communication-link with operators far away (through the CNA relaying acoustic communication and electromagnetic). This chapter shows how different path planning and knowledge about acoustic localisation methods can be combined to improve the reliability of AUVs while submerged.

Chapter 4 presents a path planner to find a path from an initial configuration q_{start} to a goal q_{goal} while taking the vehicle's motion constraints into consideration. It is an extension to HA* to handle online planning and reparation of paths in 3D. It uses priority-based expansion of a search tree which uses a motion pattern to create new configurations for expansion of the vehicle to ensure that found paths are both feasible and collision free. It has been developed for non-holonomic vehicles such as a torpedo-shaped AUV. However, the algorithm is based on a supplied pattern representing a discrete set of feasible motions, and can hence be adapted to an arbitrary vehicle.

Chapter 5 presents motion planner for leader-follower scenarios. The planner is for control of the follower. It is based on an extension of the planner presented in chapter 4. It is based on expanding a search tree using a motion pattern to minimise the average distance between the vehicles over time. The planner takes the kinematic constraints of the follower vehicle into consideration to be able to find paths which aim to minimise the average distance over time between the vehicles in scenarios where the follower vehicle's minimum speed is greater than the leader's maximum.

Chapter 6 extends chapter 5 where a surface vehicle is acting as a follower to have a surface vehicle to have it act as a navigational aid to submerged AUVs. In this chapter, instead of having the goal to reduce the distance between the vehicles, the surface vehicle instead searches for spatio-temporal paths for transmission of acoustic messages where and when they should help reduce the uncertainty of the receiving AUVs to the greatest extent. This approach uses a mixture between exploration using a search tree with sampling-based techniques to plan a list of waypoints for the vehicle based on sparse updates from the AUVs.

7.2 Contributions From the Thesis

The following section gives a brief presentation of some of the work which has been and is planned to be published from this thesis.

7.2.1 Online Path Planning Methods

This thesis has led to the development of multiple path planning methods able to produce plans for various scenarios in real-time. The first one presented in chapter 4 is an online path planner with re-planning capabilities which has not yet been published as the plan is to extend this work with anytime planning and then publish. The second planner presented is a leader-follower scenario which handles *all* types of scenarios where leader and follower are under either same or different motion capabilities. This work was presented at IEEE MTS/OCEANS 2018 in Charleston [158]. The third planner presented aims to combine knowledge on how to improve acoustic localisation and path-planning as a path-planning approach. This work plans the path of a support vehicle to transmit localisation messages to AUVs from locations, and times, at which the reduction of uncertainty on the receiving platforms should be the greatest. This work was published as an IEEE Robotics and Automation Letter [159] and will be presented at IEEE/RSJ International Conference on Intelligent Robots and Systems in Macau, 2019.

These three planners go from the classical objective of planning a path from an initial state to a goal region to more application based path planning. The first presented planner is extended to the second which instead of reaching a goal, has as objective the minimisation of distance between two vehicles. Which in turn lead to the work of planning a path for a vehicle aiming to stay within communication distance and to improve geometric relationships for acoustic localisation.

7.2.2 Acoustic Localisation

The work of this thesis started with looking at acoustic localisation in cooperative missions between Autonomous Surface Vehicles and Autonomous Underwater Vehicles, where the first project led to the development of a Moving Long Baseline simulator for Non-linear Least Squares using multiple vehicles. A paper for this was published in IEEE MTS/OCEANS 2017, Aberdeen [160]. It later changed direction more towards cooperative operations. This included a spatio-temporal planner from transmitting localisation messages. A part of that problem was to look at when to transmit localisation messages, within a Time-Division Multiple Access time slot, such that the uncertainty on receiving vehicles are estimated to be reduce by the most. A paper on this was presented at IEEE MTS/OCEANS in Marsielle 2019 [159], this work has been described in section 5.4.

7.2.3 Communication Relay

During this thesis collaborations with a group working on trying to improve the confidence and situation understanding for AUVs operators has been performed. Their research is on how to use natural language to present the very limited data transmitted from Autonomous Underwater Vehicles over acoustic communication to a more understandable format, in this case a chat-bot[1]. For some missions performed using of the shelf products (Iver-2, Sonobot and SeeByte software) they needed a way to listen to the acoustic messages from the Autonomous Underwater Vehicles while not necessarily being at the location of the operation. From this requirement a communication-relay framework was developed to use a surface vehicle as a translator from acoustic communication to wi-fi, enabling the operators to chat with the submerged vehicle using a surface vehicle with multi-modal communication. This communication relay software was part of a publication in IEEE OES Autonomous Underwater Vehicle Symposium 2018 in Porto [162], and the 2019 ACM conference on Designing Interactive Systems in San Diego [163].

[1]MIRIAM [161]

7.3 Future Work

This thesis is not an absolute solution to any of the problems it addresses. It does however present novel work for path planning which has shown good results for the specified scenarios. However, there are still many exciting opportunities to extend the work here to improve it even further.

Anytime Dynamic Hybrid-State A*. The current state of PBHA* includes re-planning of paths using previously explored branches in the search tree. As was shown in chapter 4, using highly weighted heuristics can find sub-optimal paths in a fraction of the time as non-weighted heuristics. As the weight goes towards 1, the solution goes towards the optimal as well as the number of expanded node increases (and therefore also the time). By initially solving the problem with highly weighted heuristics and starting to execute the plan, the vehicle can during its path keep decreasing the weight while updating and performing the search from the current tree until either the goal is found or the weight becomes 1 (non-weighted).

Maximise optical communication time for a leader-follower scenario. Chapter 5 was developed based on the results from a trial to test optical communication between a submerged and a surface vehicle. The surface vehicle was roughly at all time 3 times faster than the AUV and hence could not stay on top of it for a long time. If the region in which the optical communication can be used is known, it could be used as an objective for the planner to aim to maximise the time where this method of high bandwidth communication can be used.

Maximise FIM as path planning goal for improved NLS acoustic localisation. The planner presented in chapter 6 to improve acoustic localisation on AUVs by positioning the transmitter is based on estimating the results on an EKF. However, as has been shown earlier both in this thesis and references provided, NLS is in many cases the superior range-only localisation method. By instead planning to maximise the FIM for all vehicles being supported, the same method should be usable as long as the cost function is changed. Having both cost functions available would make for a versatile algorithm able to cope with many situations. Similar thoughts can be applied to localising nodes in a sensor network.

In-water trials. Up to the point of writing this thesis there have only been limited in-water trials. There are planned integrations with SeeByte's autonomy framework Neptune [152] for some of the work in chapter 5 and 6. The leader-follower algorithm has been integrated on a Raspberry PI and shown to be able to run in real-time. However, due to the vehicle not being tested before trial with Neptune, issues occurred which could not be solved during the trials.

A Appendix

This appendix will serve as a description of relevant work for the thesis, which did not have a natural place within the main part of it.

A.1 EvoLogics Modem ROS Drivers

To enable the usage of the EvoLogics modem [164] equipped on available vehicles — a driver for ROS was developed. This was then used for sea trials both for data communication and distance measurement. The drivers were also implemented into the necessary products from SeeByte to enable the vehicles to be used with SeeTrack and Neptune as well as to enable the communication relay described below.

A.2 Communication Relay

During this thesis, a software for communication relay was developed. The implementation is done as a ROS-node [142]. It is running as a TCP server, allowing connected nodes to receive and transmit acoustic messages through the communication relay. It has been used successfully in multiple missions [162, 163], allowing a CCC placed in a position where it could not deploy an acoustic modem in the water to use an ASV (Sonobot) to relay information to and from an AUV (IVER-3). The work was integrated with the commercially available products from Seebyte: Neptune [152] and SeeTrack [165], along with the natural language interface MIRIAM [161]. This allowed operators to in a natural way query the submerged and operating AUV about its mission and vehicle status etc., without having any direct way to communicate with it.

A.3 Sonobot

EvoLogics have developed a small catamaran ASV named Sonobot [166] as can be seen in Fig. 89. The Sonobot have been used both as an autonomous vehicle and as the platform

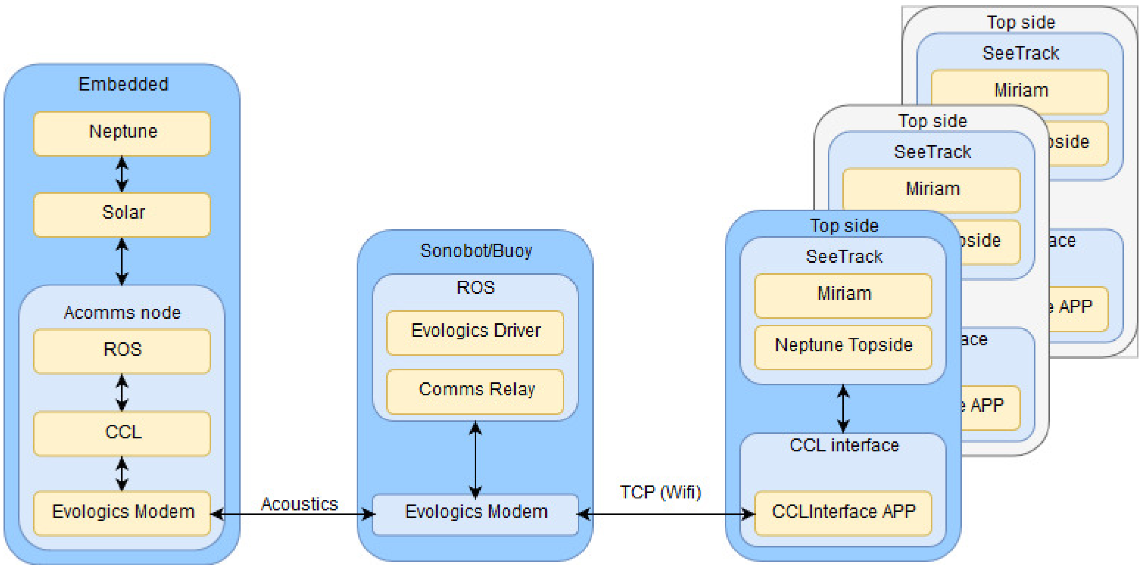


Fig. 88: A TCP server allows other platforms to receive and transmit acoustic messages through a platform equipped with an acoustic modem. Running as a stand-alone ROS-node this system can be deployed on an arbitrary vehicle to easily enable communication relay.

acting as the communication relay described.

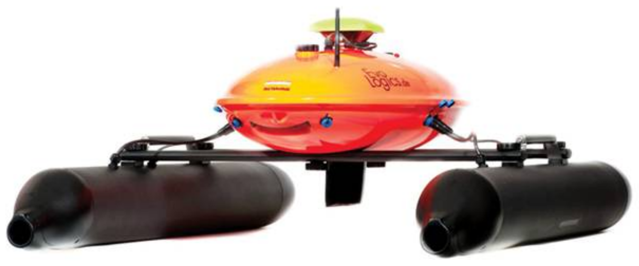


Fig. 89: Sonobot is an ASV from EvoLogics.

To enable Neptune to work on the vehicle, a backseat computer (Raspberry PI) was integrated to be able to control and monitor the robot, otherwise running DUNE, through Neptune as a backseat driver.

Bibliography

[1] A. Cormack, D. M. Lane, and J. Wood, "Operational Experiences for Maritime Homeland Security Operations," *OCEANS'10 IEEE SYDNEY*, pp. 1–7, 2010.

[2] S. Reed, Y. Petillot, and J. Bell, "An automatic approach to the detection and extraction of mine features in sidescan sonar," *IEEE Journal of Oceanic Engineering*, vol. 28, no. 1, pp. 90–105, 2003.

[3] P. Ridao, M. Carreras, and D. e. Ribas, "Intervention AUVs: The Next Challenge," *19th World Congress of The International Federation of Automatic Control*, p. 12146, 2014.

[4] D. Bingham, T. Drake, A. Hill, R. Lott, and A. W. Hill, "The Application of Autonomous Underwater Vehicle (AUV) Technology in the Oil Industry – Vision and Experiences," *TS4.4 Hydrographic Surveying II*, pp. 1–13, 2002.

[5] W. G. E. Group, "https://www.westwoodenergy.com/product/world-auv-market-forecast-2018-2022/," 2018.

[6] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[7] F. Dayoub, M. Dunbabin, and P. Corke, "Robotic detection and tracking of Crown-of-Thorns starfish," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 1921–1928, 2015.

[8] C. Kunz, C. Murphy, R. Camilli, H. Singh, J. Bailey, R. Eustice, M. Jakuba, K. I. Nakamura, C. Roman, T. Sato, R. A. Sohn, and C. Willis, "Deep sea underwater robotic exploration in the ice-covered arctic ocean with AUVs," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 3654–3660, 2008.

[9] C. Roman and R. Mather, "Autonomous underwater vehicles as tools for deep-submergence archaeology," *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 224, no. 4, pp. 327–340, 2010.

[10] A. Vasiljević, D. Nac, N. Stilinović, N. Mišković, and Z. Vukić, “Application of an ASV for Coastal Underwater Archaeology,” *Pomorski zbornik*, vol. Posebno iz, no. 1, pp. 179–186, 2016.

[11] B. Li, B. Moridian, and N. Mahmoudian, “Underwater multi-robot persistent area coverage mission planning,” *OCEANS 2016 MTS/IEEE Monterey, OCE 2016*, pp. 1–6, 2016.

[12] R. R. Murphy, “A decade of rescue robots,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 5448–5449, 2012.

[13] L. D. Stone, C. M. Keller, T. M. Kratzke, and J. P. Strumpfer, “Search for the wreckage of Air France Flight AF 447,” *Statistical Science*, vol. 29, no. 1, pp. 69–80, 2014.

[14] L. Paull, S. Saeedi, M. Seto, and H. Li, “AUV navigation and localization: A review,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[15] I. Mahon and S. Williams, “SLAM using Natural Features in an Underwater Environment,” *International Conference on Control, Automation, Robotics and Vision*, 2004.

[16] I. Mahon, S. B. Williams, O. Pizarro, and M. Johnson-Roberson, “Efficient view-based SLAM using visual loop closures,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1002–1014, 2008.

[17] Y. Petillot, I. Tena Ruiz, and D. M. Lane, “Underwater vehicle path planning using a multi-beam forward looking sonar,” *IEEE Journal of Oceanic Engineering*, vol. 26, no. 2, pp. 240–251, 2001.

[18] S. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte, “Autonomous underwater simultaneous localisation and map building,” *Update*, no. April 2000, pp. 1793–1798, 2002.

[19] E. Olson, J. J. Leonard, and S. Teller, “Robust range-only beacon localization,” *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2006.

[20] J. Vaganay, J. J. Leonard, J. Curcio, and J. Willcox, “Experimental validation of the moving long base-line navigation concept,” *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*, no. l, pp. 59–65, 2004.

[21] A. Bahr and J. J. Leonard, “Minimizing trilateration errors in the presence of uncertain landmark positions,” *Proc. 3rd European Conf. on Mobile Robots (ECMR)*, pp. 48–53, 2007.

[22] M. F. Fallon, G. Papadopoulos, J. J. Leonard, and N. M. Patrikalakis, "Cooperative AUV navigation using a single maneuvering surface craft," *International Journal of Robotics Research*, vol. 29, no. 12, pp. 1461–1474, 2010.

[23] S. E. Webster, J. M. Walls, L. L. Whitcomb, and R. M. Eustice, "Decentralized extended information filter for single-beacon cooperative acoustic navigation: Theory and experiments," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 957–974, 2013.

[24] F. Mandi, "Underwater navigation and localization using single range measurements,"

[25] M. S. Wiig, T. R. Krogstad, and O. Midtgaard, "Autonomous identification planning for mine countermeasures," *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*, 2012.

[26] M. F. Fallon, "Simultaneous Localization and Mapping in Marine Environments.," *Marine Robot Autonomy*, pp. 329–372, 2013.

[27] Y. Pailhas, P. Patron, J. Cartwright, F. Maurelli, J. Sawas, Y. Petillot, N. Valeyrie, and R. Campus, "Fully Integrated Multi-Vehicles Mcm," *International Conference and Exhibition on Underwater Acoustic Measurements: Technologies & Results*, 2011.

[28] A. Kelly, "Precision dilution in triangulation based mobile robot position estimation," *In Intelligent Autonomous Systems*, 2003.

[29] A. Bahr, J. J. Leonard, and A. Martinoli, "Dynamic positioning of beacon vehicles for cooperative underwater navigation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3760–3767, 2012.

[30] M. Stojanovic, J. Preisig, and W. Hole, "Underwater Acoustic Communication Channels: Propagation Models and Statistical Characterization," *IEEE Communications Magazine*, no. January, pp. 84–89, 2009.

[31] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*. No. 1, Springer, 2:nd ed., 2011.

[32] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–108, 2006.

[33] J. Jung, Y. Lee, D. Kim, D. Lee, H. Myung, and H. T. Choi, "AUV SLAM using forward/downward looking cameras and artificial landmarks," *2017 IEEE OES International Symposium on Underwater Technology, UT 2017*, pp. 1–3, 2017.

[34] B. He, Y. Liang, X. Feng, R. Nian, T. Yan, M. Li, and S. Zhang, "AUV SLAM and experiments using a mechanical scanning forward-looking sonar," *Sensors (Switzerland)*, vol. 12, no. 7, pp. 9386–9410, 2012.

[35] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Bathymetric SLAM with no map overlap using Gaussian Processes," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1242–1248, 2011.

[36] A. Palomer, "3D Underwater SLAM Using Sonar and Laser Sensors," *Doctoral Thesis*, 2018.

[37] G. Salavasidis, A. Munafò, C. A. Harris, T. Prampart, R. Templeton, M. Smart, D. T. Roper, M. Pebody, S. D. McPhail, E. Rogers, and A. B. Phillips, "Terrain-aided navigation for long-endurance and deep-rated autonomous underwater vehicles," *Journal of Field Robotics*, vol. 36, no. 2, pp. 447–474, 2019.

[38] C. M. G. Gussen, P. S. R. Diniz, M. L. R. de Campos, W. A. Martins, F. M. Costa, and J. N. Gois, "A Survey of Underwater Wireless Communication Technologies," *Journal of Communication and Information Systems*, vol. 31, no. 1, pp. 242–255, 2016.

[39] M. Chaplin, "http://www1.lsbu.ac.uk/water/water_vibrational_spectrum.html."

[40] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, p. 34, 2008.

[41] S. Jamshidi and M. N. A. Bakar, "An Analysis on Sound Speed in Seawater using CTD Data," *Journal of Applied Sciences*, no. 10, pp. 132–138, 2010.

[42] Webpage, "Scientists, Federation Of American: https://fas.org/man/dod-101/navy/docs/es310/SNR_PROP/snr_prop.htm."

[43] B. Jin, X. Xu, and T. Zhang, "Robust time-difference-of-arrival (Tdoa) localization using weighted least squares with cone tangent plane constraint," *Sensors (Switzerland)*, vol. 18, no. 3, 2018.

[44] J. C. Kinsey, D. A. Smallwood, and L. L. Whitcomb, "A new hydrodynamics test facility for UUV dynamics and control research," *Oceans 2003: Celebrating the Past... Teaming Toward the Future*, vol. 1, no. March, pp. 356–361, 2003.

[45] M. M. Hunt, W. M. Marquet, D. A. Moller, K. R. Peal, W. K. Smith, and R. C. Spindel, "An acoustic navigation system," *Technical Report WHOI- 74-6*, 1974.

[46] D. Carta, "Optimal Estimation of Undersea Acoustic Transponder Locations," *Oceans*, pp. 466–471, 1978.

[47] N. H. Kussat, C. D. Chadwell, and R. Zimmerman, “Absolute positioning of an autonomous underwater vehicle using GPS and acoustic measurements,” *IEEE Journal of Oceanic Engineering*, vol. 30, no. 1, pp. 153–164, 2005.

[48] H. Thomas, “GIB buoys: an interface between space and depths of the oceans,” *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles*, pp. 181–184, 2002.

[49] A. Alcocer, P. Oliveira, and A. M. Pascoal, “Underwater Acoustic Positioning Systems Based on Buoys with GPS,” *In Proc. of the 8th European Conference on Underwater Acoustics ECUA*, no. January, 2006.

[50] P. Ridao, D. Ribas, E. Hernàndez, and A. Rusu, “USBL/DVL navigation through delayed position fixes,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2344–2349, 2011.

[51] P. Rigby, O. Pizarro, and S. B. Williams, “Towards Geo-Referenced Fusion of Navigation Through DVL Measurements,” *Oceans*, no. 2, 2006.

[52] P. Ridao, M. Carreras, D. Ribas, and R. Garcia, “Visual inspection of hydroelectric dams using an autonomous underwater vehicle,” *Journal of Field Robotics*, vol. 27, no. 6, pp. 759–778, 2010.

[53] D. Heckman and R. Abbott, “An acoustic navigation technique,” *Engineering in the Ocean Environment, Ocean 73 - IEEE International Conference on*, vol. 3, no. 1, pp. 591–595, 1973.

[54] T. Zhang, L. Chen, and Y. Li, “AUV underwater positioning algorithm based on interactive assistance of SINS and LBL,” *Sensors (Switzerland)*, vol. 16, no. 1, 2016.

[55] A. Alcocer, “Positioning and Navigation Systems for Robotic Underwater Vehicles,” *PhD Thesis*, no. January 2010, 2010.

[56] J. Curcio, J. J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, and M. Grund, “Experiments in moving baseline navigation using autonomous surface craft,” *Proceedings of MTS/IEEE OCEANS, 2005*, vol. 2005, 2005.

[57] Y. T. Tan, R. Gao, and M. Chitre, “Cooperative path planning for range-only localization using a single moving beacon,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 371–385, 2014.

[58] P. Baccou and B. Jouvencel, “Homing and navigation using one transponder for AUV, postprocessing comparisons results with long base-line navigation,” *Proceedings 2002 IEEE International Conference on Robotics and Automation*, no. May, pp. 4004–4009, 2002.

[59] D. Ribas, P. Ridao, A. Mallios, and N. Palomeras, "Delayed state information filter for USBL-Aided AUV navigation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4898–4903, 2012.

[60] O. Hegrenas, K. Gade, O. Hagen, and P. Hagen, "Underwater transponder positioning and navigation of autonomous underwater vehicles," *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pp. 1–7, 2009.

[61] R. M. Eustice, L. L. Whitcomb, H. Singh, and M. Grund, "Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4257–4264, 2007.

[62] G. Cario, A. Casavola, V. Djapic, P. Gjanci, M. Lupia, C. Petrioli, and D. Spaccini, "Clock synchronization and ranging estimation for control and cooperation of multiple UUVs," *OCEANS 2016 - Shanghai*, 2016.

[63] A. T. Gardner and J. A. Collins, "Advancements in High-Performance Timing for Long Term Underwater Experiments," *2012 Oceans*, pp. 1–8, 2012.

[64] R. Almeida, J. Melo, and N. Cruz, "Characterization of measurement errors in a LBL positioning system," *OCEANS 2016 - Shanghai*, 2016.

[65] R. N. Jazar, *Theory of applied robotics: Kinematics, dynamics, and control (2nd Edition)*. Springer, secibd ed., 2010.

[66] G. Papadopoulos, M. F. Fallon, J. J. Leonard, and N. M. Patrikalakis, "Cooperative localization of marine vehicles using nonlinear state estimation," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 4874–4879, 2010.

[67] S. Alakkari, K. Jamil, and S. Alhumaidi, "Range only target localization in multistatic passive radar system: A gradient descent approach," *2015 Signal Processing Symposium, SPSympo 2015*, no. 2, pp. 1–4, 2015.

[68] N. Loizou and P. Richtárik, "Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods," *ArXiv*, 2017.

[69] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[70] W. Chen, W. Yan, R. Cui, and H. Cui, "Optimal configuration of USVs for Moving Long Baseline positioning system," *ICARM 2016 - 2016 International Conference on Advanced Robotics and Mechatronics*, no. 2, pp. 394–398, 2016.

[71] A. Munafo, J. Sliwka, and J. Alves, "Dynamic placement of a constellation of surface buoys for enhanced underwater positioning," *MTS/IEEE OCEANS 2015 - Genova: Discovering Sustainable Ocean Energy for a New World*, pp. 1–6, 2015.

[72] T. Glotzbach, D. Moreno-Salinas, A. Pascoal, and J. Aranda, "Optimal sensor placement for acoustic range-based underwater robot positioning," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 9, no. PART 1, pp. 215–220, 2013.

[73] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960.

[74] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, p. 182, 1997.

[75] G. Antonelli, F. Arrichiello, S. Chiaverini, and G. S. Sukhatme, "Observability analysis of relative localization for AUVs based on ranging and depth measurements," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4276–4281, 2010.

[76] R. Gao and M. Chitre, "Cooperative positioning using range-only measurements between two AUVs," *OCEANS'10 IEEE Sydney, OCEANSSYD 2010*, pp. 1–6, 2010.

[77] P. Del Moral, "Nonlinear filtering: Interacting particle resolution," *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, vol. 325, no. 6, pp. 653–658, 1996.

[78] J. M. Hammersley and K. W. Morton, "Poor Man's Monte Carlo," *Journal of the Royal Statistical Society*, vol. 16, no. 1, pp. 23–38, 1954.

[79] W. Hereman and S. W. Murphy, "Determination of a Position in Three Dimensions Using Trilateration and Approcimate Distances."

[80] L. Wang and S. Pang, "AUV Navigation Based on Inertial Navigation and Acoustic Positioning Systems," *OCEANS 2018 MTS/IEEE Charleston*, pp. 1–8, 2018.

[81] J. Vaganay, J. Leonard, and J. Bellingham, "Outlier rejection for autonomous acoustic navigation," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2174–2181, 1996.

[82] Australian Centre for Field Robotics, "AUV Sirius, http://marine.acfr.usyd.edu.au/systems/auv-sirius/."

[83] C. R. German, M. V. Jakuba, J. C. Kinsey, J. Partan, S. Suman, A. Belani, and D. R. Yoerger, "A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of Autonomous Surface Vehicles

and Autonomous Underwater Vehicles," *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*, pp. 1–7, 2012.

[84] E. I. Sarda and M. R. Dhanak, "A USV-Based Automated Launch and Recovery System for AUVs," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 1, pp. 37–55, 2017.

[85] E. I. Sarda and M. R. Dhanak, "Launch and Recovery of an Autonomous Underwater Vehicle from a Station-Keeping Unmanned Surface Vehicle," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 290–299, 2019.

[86] V. H. Pinto, N. A. Cruz, R. M. Almeida, and C. F. Gonccalves, "ALARS - Automated Launch and Recovery System for AUVs," *OCEANS 2018 MTS/IEEE Charleston, OCEAN 2018*, 2018.

[87] A. Bahr and J. J. Leonard, "Cooperative Localization for Autonomous Underwater Vehicles," *Experimental Robotics*, no. Dvl, pp. 387–395, 2008.

[88] M. Chitre, "Path planning for cooperative underwater range-only navigation using a single beacon," *IEEE 2010 International Conference on Autonomous and Intelligent Systems, AIS 2010*, pp. 1–6, 2010.

[89] T. Y. Teck and M. Chitre, "Single beacon cooperative path planning using Cross-Entropy method," *Oceans 2011*, pp. 1–6, 2011.

[90] J. D. Quenzer and K. A. Morgansen, "Observability based control in range-only underwater vehicle localization," *Proceedings of the American Control Conference*, pp. 4702–4707, 2014.

[91] W. Yan, W. Chen, R. Cui, and H. Li, "Optimal distance between mobile buoy and target for moving long baseline positioning system," *Journal of Navigation*, vol. 68, no. 4, pp. 809–826, 2015.

[92] B. Englot and F. S. Hover, "Sampling-based coverage path planning for inspection of complex structures," *22nd International Conference on Automated Planning and Scheduling, ICAPS 2012*, pp. 29–37, 2012.

[93] E. Galceran, "Coverage Path Planning for Autonomous Underwater Vehicles," *Doctoral Thesis*, 2014.

[94] A. Matos and N. Cruz, "Coordinated operation of autonomous underwater and surface vehicles," *Oceans Conference Record (IEEE)*, pp. 2–7, 2007.

[95] P. Švec, B. C. Shah, I. R. Bertaska, W. Klinger, A. J. Sinisterra, K. Von Ellenrieder, M. Dhanak, and S. K. Gupta, "Adaptive Sampling Based COLREGs-Compliant Obstacle Avoidance for Autonomous Surface Vehicles," *Ieee*, 2014.

[96] M. Brand, M. Masuda, N. Wehner, and X.-H. Yu, “Ant Colony Optimization Algorithm for Robot Path Planning,” *International Conference On Computer Design And Appliations*, vol. 3, pp. 436–440, 2010.

[97] L. V. J. and S. A. A., “Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape,” *Algorithmica*, pp. 403–430, 1987.

[98] C. Petres, Y. Pailhas, Y. Petillot, and D. Lane, “Underwater path planing using fast marching algorithms,” *Europe Oceans 2005*, vol. 2, pp. 814 – 819, 2005.

[99] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, “Path planning for autonomous underwater vehicles,” *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 331–341, 2007.

[100] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, “Principles of Robot Motion Theory, Algorithms, and Implementations,” *MIT press*, 2005.

[101] S. M. LaValle, *Planning Algorithms.* New York, NY, USA: Cambridge University Press, 2006.

[102] T. Lozano-Pérez, “Spatial Planning: A Configuration Space Approach,” *IEEE Transactions on Computers*, vol. C-32, no. 2, pp. 108–120, 1983.

[103] G. Ausiello, V. Bonifaci, and L. Laura, “The on-line asymmetric traveling salesman problem,” *Journal of Discrete Algorithms*, vol. 6, no. 2, pp. 290–298, 2008.

[104] K. Carroll, E. Nelson, S. McClaran, G. William, D. Friesen, and D. Barnett, “AUV path planning: an A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones,” *Autonomous Underwater Vehicle Technology (AUV)*, 1992.

[105] E. W. Dijkstra, “A note on two problems in connection with graphs,” 1959.

[106] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *Systems Science and Cybernetics*, no. 2, pp. 100–107, 1968.

[107] J. Barraquand, L. Kavraki, J. C. Latombe, R. Motwani, T. Y. Li, and P. Raghavan, “A random sampling scheme for path planning,” *International Journal of Robotics Research*, vol. 16, no. 6, pp. 759–774, 1997.

[108] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensionalconfiguration spaces,” *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566 – 580, 1996.

[109] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Report No. TR 98-11*, 1998.

[110] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[111] S. Karaman and E. Frazzoli, "Incremental Sampling-based Algorithms for Optimal Motion Planning," *Robotics: Science and Systems (RSS)*, 2010.

[112] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2997–3004, 2014.

[113] D. Ferguson and A. Stentz, "Field D*: An Interpolation-Based Path Planner and Replanner," *Robotics Research*, pp. 239–253, 2007.

[114] N. D. Richards and M. Sharma, "A Hybrid A*/Automaton Approach to on-Line Path Planning with Obstacle Avoidance," *AIAA*, pp. 141–157, 2004.

[115] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical Search Techniques in Path Planning for Autonomous Driving," *First International Symposium on Search Techniques in Aritficial Interlligence and Robotics*, 2008.

[116] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[117] J. Petereit, T. Emter, C. W. Frey, T. Kopfstedt, and A. Beutel, "Application of Hybrid A* to an Autonomous Mobile Robot for Path Planning in Unstructured Outdoor Environments," *ROBOTIK 2012; 7th German Conference on Robotics*, no. 1, pp. 227–232, 2012.

[118] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1478–1483, 2011.

[119] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments," *IEEE International Conference on Robotics and Automation*, no. May, 1994.

[120] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain," *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 1, no. May, pp. 968–975, 2003.

[121] S. Koenig and M. Likhachev, "Incremental A*," *Nips*, pp. 1539–1546, 2001.

[122] M. Pivtoraiko and A. Kelly, "Generating near minimal spanning control sets for constrained motion planning in discrete state spaces," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 594–600, 2005.

[123] A. Kelly, O. Amidi, M. Bode, M. Happold, H. Herman, T. Pilarski, P. Rander, A. Stentz, N. Vallidis, and R. Warner, "Toward reliable off road autonomous vehicles operating in challenging environments," *Springer Tracts in Advanced Robotics*, vol. 21, no. 5, pp. 599–608, 2006.

[124] Y. J. Heo and W. K. Chung, "RRT-based path planning with kinematic constraints of AUV in underwater structured environment," *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2013*, pp. 523–525, 2013.

[125] D. Hsu, "Randomized Single-query Motion Planning in Expansive Spaces," *Doctoral Thesis*, no. May, p. 118, 2000.

[126] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[127] D. HSU, J.-C. LATOMBE, and R. MOTWANI, "Path Planning in Expansive Configuration Spaces," *International Journal of Computational Geometry & Applications*, vol. 09, no. 04n05, pp. 495–512, 2003.

[128] J. M. Phillips, N. Bedrossian, and L. E. Kavraki, "Guided expansive spaces trees: A search strategy for motion- and cost-constrained state spaces," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 4, pp. 3968–3973, 2004.

[129] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Discrete search leading continuous exploration for kinodynamic motion planning," *Robotics: Science and Systems*, vol. 3, pp. 313–320, 2008.

[130] E. Vidal, M. Moll, N. Palomeras, J. D. Hernández, M. Carreras, and L. E. Kavraki, "Online Multilayered Motion Planning with Dynamic Constraints for Autonomous Underwater Vehicles," *IEEE International Conference on Robotics and Automation (ICRA)*, p. (to appear), 2019.

[131] J. D. Hernández, M. Moll, E. Vidal, M. Carreras, and L. E. Kavraki, "Planning feasible and safe paths online for autonomous underwater vehicles in unknown environments," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 1313–1320, 2016.

[132] L. E. Dubins, “On Curves of Minimal Length with a Constraint on Average Curvature , and with Prescribed Initial and Terminal Positions and Tangents,” *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[133] Y. H. Wang, W. Y. Cai, and Y. R. Zheng, “Dubins Curves for 3D Multi-Vehicle Path Planning Using Spline Interpolation,” *OCEANS 2017 MTS/IEEE Anchorage*, pp. 1–5, 2017.

[134] W. Cai and M. Zhang, “Smooth 3D dubins curves based mobile data gathering in sparse underwater sensor networks,” *Sensors*, vol. 18, no. 7, pp. 1–18, 2018.

[135] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[136] j. d. h. Vega, “Online Path Planning for Autonomous Underwater Vehicles Under Motion Constraints,” *Doctoral Thesis*, 2017.

[137] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[138] O. Brock and O. Khatib, “High-speed navigation using the global dynamic window approach,” *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 1, no. 1, pp. 341–346, 1999.

[139] B. O. H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig, “A modified dynamic window algorithm for horizontal collision avoidance for AUVs,” *2016 IEEE Conference on Control Applications, CCA 2016*, pp. 499–506, 2016.

[140] I. Tusseyeva, S.-G. Kim, and Y.-G. Kim, “3D Global Dynamic Window Approach for Navigation of Autonomous Underwater Vehicles,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 13, no. 2, pp. 91–99, 2013.

[141] C. J. McFarland, M. V. Jakuba, S. Suman, J. C. Kinsey, and L. L. Whitcomb, “Toward ice-relative navigation of underwater robotic vehicles under moving sea ice: Experimental evaluation in the Arctic sea,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 1527–1534, 2015.

[142] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, “ROS: an open-source Robot Operating System,” *Icra*, vol. 3, no. Figure 1, p. 5, 2009.

[143] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, 2013.

[144] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3859–3866, 2012.

[145] J. D. Hernández, M. Moll, N. Palomeras, M. Carreras, L. E. Kavraki, and E. Vidal, "Online motion planning for unexplored underwater environments using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 36, no. 2, pp. 370–396, 2018.

[146] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.

[147] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Longman Publishing Co., Inc., 1984.

[148] K. E. Bekris and L. E. Kavraki, "Greedy but safe replanning under kinodynamic constraints," *Proceedings - IEEE International Conference on Robotics and Automation*, no. April, pp. 704–710, 2007.

[149] J. Melo and A. Matos, "Guidance and control of an ASV in AUV tracking operations," *Oceans 2008*, 2008.

[150] M. Bibuli, G. Bruzzone, M. Caccia, L. Lapierre, M. Bibuli, G. Bruzzone, M. Caccia, L. L. P.-f. Algorithms, and L. Lapierre, "Path-Following Algorithms and Experiments for an Unmanned Surface Vehicle," *Journal of Field Robotics*, 2008.

[151] M. Bibuli, O. Parodi, L. Lapierre, G. Bruzzone, and M. Caccia, "Vehicle-following guidance for unmanned marine vehicles," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, no. 18, pp. 103–108, 2009.

[152] SeeByte, "http://www.seebyte.com/oceanography/neptune/."

[153] A. Munoz and A. Drogoul, "Sharing a charging station in collective robotics," *n/a*, pp. 1–9, 2002.

[154] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 631–640, 2010.

[155] J. M. Walls and R. M. Eustice, "Toward informative planning for cooperative underwater localization," *2014 Oceans - St. John's, OCEANS 2014*, pp. 1–7, 2015.

[156] J. Hudson and M. L. Seto, "Underway path-planning for an unmanned surface vehicle performing cooperative navigation for UUVs at varying depths," *IROS*, pp. 2298–2305, 2014.

[157] H. D. and S. Cohn-Vossen, “Geometry and the imagination,” *AMS Chelsea Publishing*, pp. 32–39, 1990.

[158] J. S. Willners, Y. Petillot, P. Patron, and D. Gonzalez-Adell, “Autonomous Kinodynamic Path Planning for Following and Tracking Vehicles,” *OCEANS 2018 - Charleston*, 2018.

[159] J. S. Willners, L. Toohey, and Y. Petillot, “Improving Acoustic Range-Only Localisation by Selection of Transmission Time,” *Oceans 2019 - Marsielle*, pp. 1–6, 2019.

[160] J. S. Willners, P. Patron, and Y. Petillot, “Moving Baseline Localization for Multi-Vehicle Maritime Operations,” *OCEANS - Aberdeen*, 2017.

[161] H. Hastie, F. J. C. Garcia, D. A. Robb, P. Patron, and A. Laskov, “MIRIAM: A Multimodal Chat-Based Interface for Autonomous Systems,” *19th ACM International Conference on Multimodal Interaction*, pp. 1–2, 2017.

[162] D. A. Robb, J. S. Willners, N. Valeyrie, F. J. C. Garcia, A. Laskov, X. Liu, P. Patron, H. Hastie, and Y. Petillot, “A Natural Language Interface with Relayed Acoustic Communications for Improved Command and Control of AUVs,” *AUV - Porto*, 2018.

[163] D. A. Robb, J. Lopes, S. Padilla, A. Laskov, F. J. C. Garcia, X. Liu, J. S. Willners, N. Valeyrie, K. Lohan, D. Lane, P. Patron, Y. Petillot, M. J. Chantler, and H. Hastie, “Exploring Interaction with Remote Autonomous Systems using Conversational Agents,” *The 2019 ACM conference on Designing Interactive Systems [DIS’19]*, 2019.

[164] EvoLogics, “https://evologics.de/acoustic-modems.”

[165] SeeByte, “http://www.seebyte.com/military/seetrack-military/.”

[166] Sonobot, “https://www.evologics.de/en/products/sonobot/index.html.”