Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Sindre Løining Skaar

Mission Planning for Wave Driven Autonomous Surface Vessels

Master's thesis in Cybernetics and Robotic
Supervisor: Tor Arne Johansen, Alberto Dallolio
November 2020

NTNU
Norwegian University of
Science and Technology

Sindre Løining Skaar

Mission Planning for Wave Driven Autonomous Surface Vessels

Master's thesis in Cybernetics and Robotic
Supervisor: Tor Arne Johansen, Alberto Dallolio
November 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

1 Abstract

Increased autonomy within the ocean vessel sector is expected to drastically change how both humans, goods and research is conducted in the coming future. Due to the increased capabilities of autonomous vehicles, they have become a more viable alternative. The vehicles have also gotten increasingly more affordable due to the reduced cost in both hardware and software. A reduction in size of many important components have also drastically increased the capabilities of smaller autonomous vessels, allowing a much broader adoption of the technology within research and the industrial sectors.

The Institute of Cybernetics and Robotics at NTNU, Trondheim is involved in the development of a broad spectre of autonomous seagoing vessels, spanning from dynamic positioning of large supply vessels to autonomous snake robots. NTNU is also involved in ocean sampling to further the understanding in aquaculture and the environmental impact of the future Norwegian development at sea. NTNU is developing self sufficient autonomous vessels capable of performing missions previously done by large and costly vessels closer to 400 tons. Small autonomous vessels still have problems navigating and staying safe in changing weather conditions. Manually planning missions often creates unfeasible missions not possible for the autonomous vessels to conduct. Too strong weather can displace small autonomous vessels hundreds of kilometres off course, leading to costly rescue missions or loss of the vessel.

This thesis has focused on increasing the capabilities of smaller autonomous vessels and reducing the chance of the vessel being carried off course. A mission planner has been developed that plans a vessel path and sensor sampling to take into account challenging weather. This allows the algorithm to create a path feasible for the vessel to conduct, while also optimising the monetary cost of the mission. To be able to predict feasible paths, the thesis has focused on finding a model for wave propelled surface vehicles to be able to better predict vessel dynamics and take account for how weather affects vessel movement. This model was then used in conjunction with a custom binary-continuous particle swarm optimisation algorithm to optimise the total estimated mission cost.

The model was tested and fitted to real life testing of a wave powered vessel called AutoNaut. Parameter estimations were conducted for both on shore locations shielded from the off shore environment and off shore environments to get a better understanding of model parameter validity.

Using the optimisation algorithm, the system was able to find feasible optimised paths where the manually created paths would not have been feasible, therefore drastically improving the vessel capabilities even in environments normally deemed too challenging for the vessel to complete.

[This page is intentionally left blank]

2 Sammendrag

En økt bruk av autonome sjøfarende fartøy er forventet å drastisk forandre hvordan både mennesker, gods og forskning blir håndtert i fremtiden. Den økte kapabiliteten til autonome fartøy har gjort dem til et mer aktuelt alternativ. Fartøyene har også blitt rimeligere grunnet en reduksjon i pris på maskinvare og programvare. En reduksjon i størrelsen på mange viktige komponenter har også drastisk økt kapabilitetene til mindre fartøy, som tillater fartøyene å kunne bli brukt i en mye større skala innenfor både forskning og industri.

Institutt for Teknisk Kybernetikk hos NTNU, Trondheim, er involvert i utviklingen av et bredt spekter av fartøy fra dynamisk posisjonering av forsyningsfartøy til slangeroboter. NTNU er også involvert i forskning innen akvakultur for å øke forståelsen av hvordan norges utvikling innen havbruk og akvakultur påvirker sjøen og det biologiske mangfoldet. For å støtte forskningen utvikler NTNU selvforsynte autonome fartøy som kan utføre oppdrag som før var utført av ekspidisjonsfartøy på opptil 400 tonn. Små autonome fartøy har derimot fortsatt problemer med å navigere og holde seg unna farlige situasjoner i vanskelige værforhold. Manuelt planlagte oppdrag er ofte umulige for autonome fartøy å gjennomføre. For sterk strøm kan for eksempel føre fartøy hundrevis av kilometer ut av kurs, noe som kan føre til dyre redningsoppdrag eller tap av fartøyet.

Denne avhandlingen har fokusert på å øke kapabilitetene til mindre autonome fartøy og redusere sannsynligheten for at fartøyene ikke kan gjennomføre oppdragene sine. For å få til dette har en oppdragsplanlegger blitt utviklet som planlegger både rute og sensorbruk i hensyn til værforhold. Dette tillater algoritmen å finne en praktisk gjennomførbar rute og samtidig optimalisere den monetære kostnaden av oppdraget. For å kunne planlegge ruter til fartøyet tar avhandlingen også for seg en matematisk modell for å beskrive dynamikken til bølgedrevne fartøy. Dette gjør det mulig å kunne forutsi hvordan vær påvirker dynamikken. Denne modellen, sammen med en egenprodusert Binær-Kontinuerlig partikkel sverm optimalisering algoritme, ble brukt for å optimalisere den estimerte oppdragskostnaden.

Den matematiske modellen ble testet og tilpasset tester av et virkelig bølgedrevet fartøy kalt AutoNaut. Parameterestimeringer av modellen ble gjort både i nære kystområder i Trondheimsfjorden, skjermet for tungt vær, og ute i åpen sjø ved Mausund.

Ved å bruke optimaliseringsalgoritmen, klarte algoritmen å finne gjennomførbare ruter hvor den manuelle metoden ikke ville være gjennomførbar. Dette øker kapabiliteten under forhold som tidligere var for utfordrende.

[This page is intentionally left blank]

3 Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Tor Arne Johansen, for giving me the opportunity to explore a field I am very passionate about and giving me the opportunity to help out in testing the AutoNaut in the sea trails. He has also been a great help in verifying and discussing different approaches and methods for conducting the master project. I would also like to thank my co supervisor Alberto Dallolio who has helped in both planning of the thesis, giving helpful feedback and giving me a greater understanding of the inner workings and practical use of the AutoNaut vessel. Furthermore, I want to thank researchers at Norsk Institutt for Vannforskning (NIVA) and Meteorologisk Institutt for providing relevant information on prevailing ASV research and detailed weather data, respectively. I also want to thank friends and family for invaluable support. Lastly, I want to thank NTNU for the great opportunity I got to pursue an education within automation and robotics, which has been a dream come true.

[This page is intentionally left blank]

List of Figures

List of Tables

[This page is intentionally left blank]

List of Algorithms

[This page is intentionally left blank]

Contents

Part I
Background

4 Introduction

4.1 Motivation

Autonomy at sea has seen a large influx of interest following the technological advances within computational power and the reduced price of computer components together with a broadened experience in autonomous vessels. Guidance systems assisting the crew in manoeuvring and planning voyages are already a given for most modern shipping vessels today, but fully autonomous vessels have yet to be adopted. Unmanned seagoing vessels have for decades been tested and proven within the academic circle, yet has only seen limited use within industry and defence. In later years however, the use of autonomous underwater vehicles, like Hugin, has been successfully used to search and locate sea mines and lost submarines, which were not found using other alternative systems, thus showing the capabilities of autonomous vessels [1, 2].

In the later years there has been an increased interest in using autonomous seagoing vessels for long term surveillance and data collection. These long-term missions bring about new challenges for the autonomous vessels. Traditional energy sources for long term sea missions such as combustion engines or in some cases nuclear powerplants are not fit for use without some degree of human supervision. These power plants also increase the minimum size of the vessel, increasing the cost of operations. Alternative propulsion methods such as wind, wave and solar energy allows for a larger payload and increased endurance than comparable non renewable autonomous vessels of similar size. Renewable energy vessels allow for a displacement of a couple hundred litres and down to half a metre of draft. Making them capable of entering shallower waters and minimising the environmental footprint. Serving both the environment and decreasing the disturbance of sensors attached to the vessel. The use of external uncontrollable energy sources such as weather does however reduce control authority of the vessel due to limited peak power. The limitation in peak power reduces the vessels capability of compensating for external forces working against the vessel.

Planning a long-term mission for renewable energy vessels can become difficult in certain situations. A planned path might be impossible to complete due to too strong current or headwinds. Vessels might get carried hundreds of kilometres off course or become uncontrollable given certain weather conditions. A way of solving this is to increase the capabilities by increasing complexity or size of the vessels. This will however drastically increase cost and reduce the usability of the vessel. Instead an algorithm is proposed that takes into account the weather effects on the vessel allowing small autonomous vessels to perform missions previously only being able to be performed by much larger vessels. Ideally the algorithm will be able to stay out of areas that slows down the vessel while exploiting winds and current that carries the vessel in the right direction. If an accurate model of the vessel is implemented, the algorithm should be able to optimise the path and sensor usage for the given vessel and mission. Optimising the mission around vessel limitations allows for smaller and cheaper vessels to be used while increasing the efficiency of the route. There is also a potential to insert limitations to assure that the risk of vessel loss is within a predefined safety margin. This allows the user to increase the scientific yield of a mission without increasing the size or equipment of the vessel. Such an algorithm could also be expanded to include multiple autonomous vessels cooperating on a common goal. With the increase of broadband communication, computing power and reduction of unit cost, the capabilities are expected to expand in the coming years.

4.2 Project and Context

In this master project, the student is expected to develop a mission planner for the vessel AutoNaut that is capable of optimising the research and speed of conducting a mission.

This thesis is divided into seven parts. The first part includes a theoretical background, motivation and state of the art research within autonomous surface vessels. The second part focuses on creating a model of the vessel and fitting the model to recorded data. This helps to understand how different parameters affects the vessel and increases the accuracy of the optimisation. The third part debates the use of a genetic optimisation algorithm to optimise the cost function for the vessel. Part four, five and six presents an analysis of the developed methods on both real life data and analytical scenarios to validate solutions to theoretical and real life tests. Part seven presents a discussion of the results, some final remarks, and directions for future research.

4.3 Previous Related Work

The master thesis is based on previous work done by the author in a semester project. In the previous work a proposal for ship model was created to model the novel hydrofoil propulsion used by the AutoNaut vessel. A cost function estimating the monetary mission cost was also suggested. The suggested mission cost included communication costs and risk of vessel loss, as well as shortest time optimisation. Given these models of the vessel dynamics, an estimated monetary mission cost was made dependent on duration of mission, mission goals, communication and weather. For the master project, the vessel model developed in previous work was used as a base to develop a more computationally efficient vessel model. The complexity of cost function was also reduced to make a clearly defined cost function.

The master project is a part of a project for NTNU to create their own autonomous vessels capable of performing scientific research for a long duration without human intervention. Thus, drastically reducing cost of missions and increasing the volume of scientific research. The Hull and propulsion hardware is made by AutoNaut, while all internals are designed by NTNU.

4.4 Types of Autonomous Surface Vessels

There are currently multiple autonomous surface vessels used in research at sea. Different types of ASVs are used, depending on mission goals and the working environment. Some vessels are focused on onshore operations lasting hours, while other vessels are made for offshore missions lasting multiple months. It should be noted that ”Autonomous surface vessel” and ”Unmanned surface vessel” as well as ”Autonomous Marine Vessels” tend to be used on both autonomous vessels and unmanned remotely operated vessels. By definition ”ASV” is a vessel that can operate without human intervention, while USV is an unmanned vessel but not necessarily autonomous. A remotely operated vessel could therefore be defined as an USV. Autonomous marine vessels entails both autonomous surface vessels and underwater vessels. For this master thesis ”ASV” will be used due to the autonomy of the vessel being in focus, however, this distinction is not always made.

4.4.1 Self Sustained ASVs

Refuelling and consumption of supplies becomes a bottleneck for how long a vessel can stay self sufficient. A vessel dependent on refuelling during its missions will be limited in reach as a refuelling depot has to always be withing range. To increase the operational range, normal vessels use a large energy storage. Human operated vessels can also use the large network of fuel pumps along the shore, and are therefore not limited by their fuel reserve. Small autonomous vessels can not be supplied by the current fuel network without external aid, and has a limited capacity for energy storage due to their small size and need for a large payload. Smaller vessels therefore have to use alternative methods to store and refill energy.

The main power consumption on small vessels normally comes form the propulsion actuators. A 250 kg boat needs a couple hundred watts to get sufficient control of the vessel. Other sources of power consumption is not expected to be much more than 50 Watts max[3]. Long duration ASVs therefore rely on wind and waves to create the power for propulsion, while using solar panels to supply power to vessel computers, sensors, communication and other electrical components.

The first autonomous vehicle to cross the Atlantic ocean was the sail driven autonomous vehicle Sailbouey by Offshore Sensing AS that completed the 2900 Km journey in 2018. The Sailbouey is a solar powered vessel that uses a sail to create propulsion. The transatlantic crossing from Newfoundland in Canada to the finish-line north of Ireland took the vessel 80 days to complete. This was the first autonomous vessel to cross the Atlantic, and the first vessel to complete the Micortransat Challenge[4]. Thus proving the robust capabilities of state of the art autonomous vessels.

4.4.2 Wind Powered ASVs

Wind powered ASVs use wind to create propulsion for the vessel. Normally by the help of a sail with a controllable yaw. The vessels are also equipped with a rudder to control the heading of the ASV. These ASVs act very much like normal sailboats, by using difference of pressure on the two sides of the sail and keel to create forward propulsion. It does however often differ slightly form normal sailboats that the sail is normally trimmed relative to the wind and not the boat (freely rotating). This is because of multiple reasons, but the main reason is that it simplifies control as the sail and rudder control can be mostly decoupled, and is also more robust[5]. Due to the low density of air, wind powered vessels need a large surface area to create a sufficient force. This is achieved by having a long mast. The long mast creates much torque, and is vulnerable in bad weather. Examples of vehicles that uses wind to propel themselves are Saildrone and Sailbouey. Both using sails to generate forward propulsion. The large wings needed for the vessels reduce the max payload of the vessel, which means that the vessels have to be larger. This complicates logistics and costs associated with the use of the vessels.

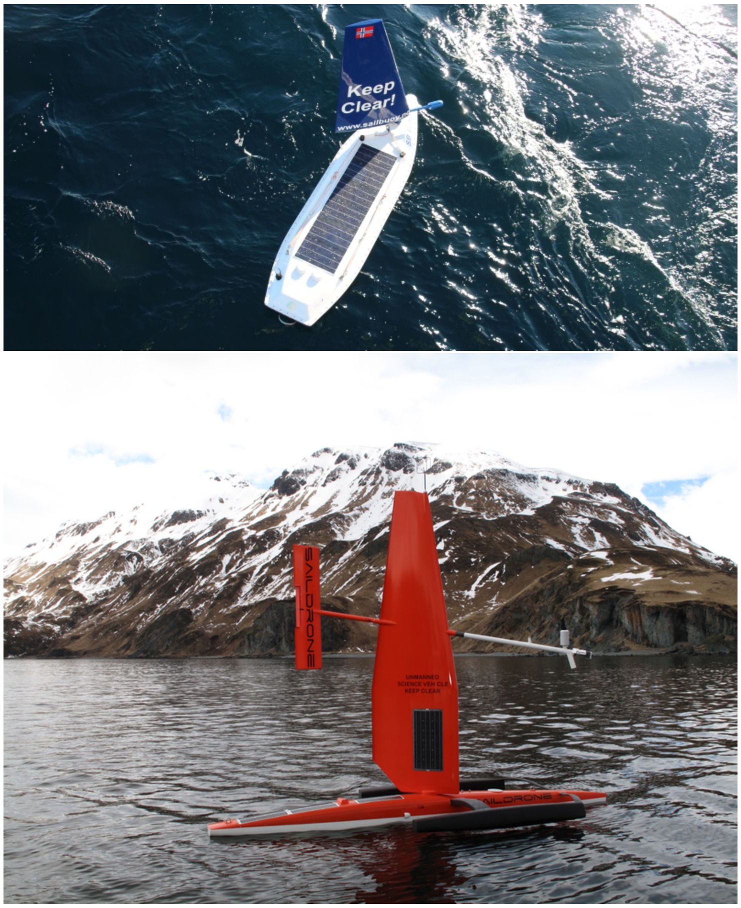


Figure 1: Examples of Wind driven Autonomous Surface vessels. Both having crossed the Atlantic without human intervention.Images: [6, 7]

4.4.3 Wave Powered ASVs

Wave powered ASVs use the change in attitude and position of the vessel created from waves to propel the ASV forward. This is normally done by making the attitude/position movement to push hydrofoils in an oscillatory motion, creating a forward thrust. There are different ways to build these hydrofoils. Two examples of hydrofoils used today are wire (WaveGlider) or rod (AutoNaut) connected hydrofoils. The density of water means that the hydrofoil surface can be drastically smaller compared to wind powered ASVs. This allows for larger payloads and easier transport. Hydrofoil ASVs do not have a large structure staying out of the water, meaning that there is no need for a large keel, and the vessel is less affected by high winds. The scalability of wave powered vessels is more difficult due to the reduction of vessel attitude rates with increased mass and inertia. Sail driven ASVs do not have the same constraints with scale. Examples of vehicles that uses waves are the wave glider and AutoNaut.

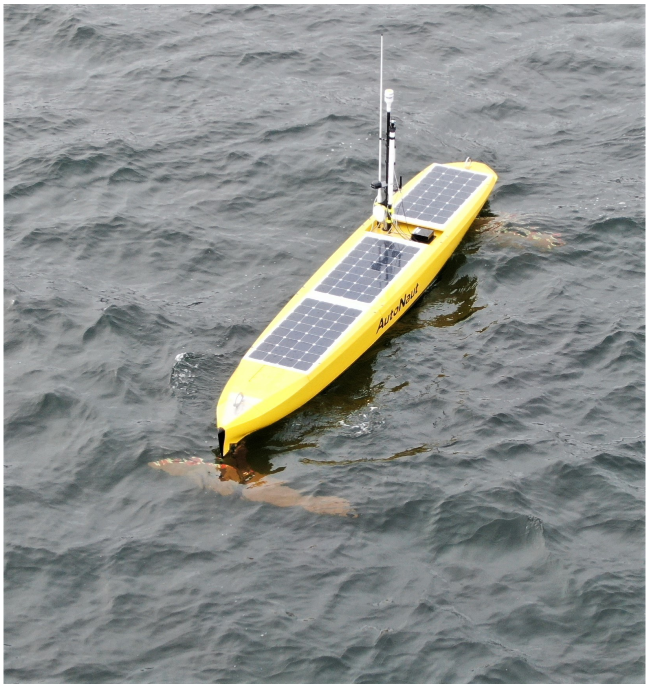


Figure 2: AutoNaut. The orange tinted surfaces observed under the AutoNaut are the hydrofoils propelling the vessel. Image: [3]

5 Methodological Approach

In order to get a thorough theoretical understanding of the scientific area relevant for the thesis, a literature review was conducted. The review included the following areas: Graph search path planning, genetic algorithms path planners, weather dynamics, linear surface wave theory and other less used algorithms tested out on ASVs. As path planning for ASVs are expected to face similar issues to AUV planners, path planning for both ASVs and AUVs has been used as a basis for the litterature review. Useful information of previous tests has been collected by discussing relevant issues with supervisors and faculty members. Additional information has been gathered from researchers at Norsk Institutt for Vannforskning (NIVA) who has implemented ASVs in their research. In addition, Meteorologisk Institutt has provided detailed weather forecasts for the algorithms. A linear 2D point model has been used to model vessel dynamics, which also has been used as a base for least squares parameter estimation for the vessel parameters. This model has been further simplified to make it work in conjunction with a particle swarm algorithm (PSO) to calculate pseudo optimal paths and sensor usage for the wave driven vessel AutoNaut.

6 State of the Art

6.1 Mission Planning

Movement and positioning of the surface vessel affects both on board sensor, sample quality and mission progression, as most constraints to a mission is position and time oriented. An optimised mission is therefore highly dependent on the vessel path. Multiple articles have mapped different approaches to optimal path planning. Multiple problem descriptions and optimisation algorithms have been proposed and implemented to solve different problems. For this master thesis, the goal is to take the optimisation one step further by implementing discrete decisions into the problem description. These discrete decisions will mainly focus on sensor sampling, but can also be extended to other areas that can implement the same problem structures.

6.1.1 Time and Energy Optimal Path Planning in General Flows

In Dhanushka Kularatne et.al. (2016) a novel method of finding time and energy optimal paths was discussed and a graph based algorithm proposed to find an optimal path between two way-points [8]. To be able to increase the efficiency in use of autonomous vessels, it is proposed that the vessel path should be dependent on vessel inertial velocity together with ocean currents, to get a better description of vessel velocity and energy expenditure. The ocean current is described as a 2D vector field, $\mathbf{v}$. The vessel velocity compared to the current is described as v_{still} and the vessel velocity to an earth fixed coordinate system is called $v_{net} = v + v_{still}$ the vessel course is described by θ. The kinematic model of the vessel is modelled as:

$$\dot{X} = v_{still}\cos\theta + v_x, \quad \dot{Y} = V_{still}\sin\theta + v_y \tag{1}$$

To simplify equations another coordinate system is made where x axis follows along current direction, and y is orthogonal to current direction. V_{net} can then be described as $[dx, dy]^T/dt$. v_{still} thus becomes:

$$||v_{still}|| = \left(\left(\frac{dx}{dt} - v\right)^2 + \left(\frac{dy}{dt}\right)^2\right)^{\frac{1}{2}} \tag{2}$$

Given a set v_{still} the duration from one point to another given constant current becomes:

$$dt = \frac{v}{v^2 v_{still}^2}dx - \frac{\sqrt{v_{still}^2(dx^2+dy^2) - v^2dy^2}}{v^2 - v_{still}^2} \tag{3}$$

A simple estimation of energy expenditure can be described by the v_{still} velocity, which describes vessel velocity relative to the ocean. Multiplying drag force times duration we get the energy expenditure. $e = \kappa ||v_{still}||^2 dt$ which becomes:

$$e = \kappa\left(\left(\frac{dx}{dt} - v\right)^2 + \left(\frac{dy}{dt}\right)^2\right)dt \tag{4}$$

Minimising energy consumption with regards to the dt parameter, we get:

$$e_{opt} = 2\kappa v(\sqrt{dx^2 + dy^2} - dx) \tag{5}$$

The model of the area is turned into a graph g, where each node on the graph represents a point on a 2D surface grid covering the area of operations. The cost of the edges between two different nodes is described by the energy and time equations. An A* search algorithm is then used to describe the cost from a starting position to the end goal.

The results in the article showed how the algorithm implemented managed to take advantage of ocean currents to optimise the path from a start point to an end point. Using a graph to describe the optimisation problem also simplifies collision avoidance in the optimisation algorithm, by not including any vertexes over land. To avoid any edges crossing land, one can dilate the surface areas or erode ocean maps and reducing the neighbour hood of vertexes. The created paths were also tested by using small vessels trans versing currents similar to the analytical currents tested. The small vessels were able to follow the preplanned path.

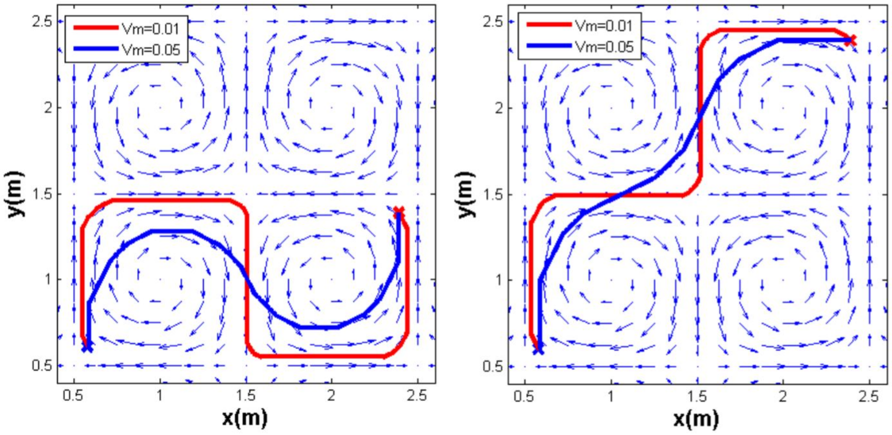


Figure 3: Image showing path results for A* algorithm. Image: [8]

6.1.2 Time Varying Flows

Most optimisation algorithms discretises the optimisation algorithm into discrete steps either time wise, position wise or both. To be able to accurately represent a continuous system, it is normal to include assumptions for the system dynamics between the time steps. Examples of such assumptions can be constant speed or non changing weather between the time steps. The accuracy of these assumptions greatly depend on the systems discretised. In an article by Dhanushka Kularatne et.al. (2017) the previous path optimisation method was extended to include time varying flows into the optimisation [9]. This is necessary for autonomous vessels at sea, often spending months at a time in the ocean. To try to minimise the error from assuming constant current velocity, the time step between each position was automatically adjusted depending on current gradient to keep the estimation error within a certain bound. Areas where the current had a large gradient, the time step was reduced, by only allowing neighbours close to the node during graph search. At areas with less gradient the neighbours were allowed to be further away.

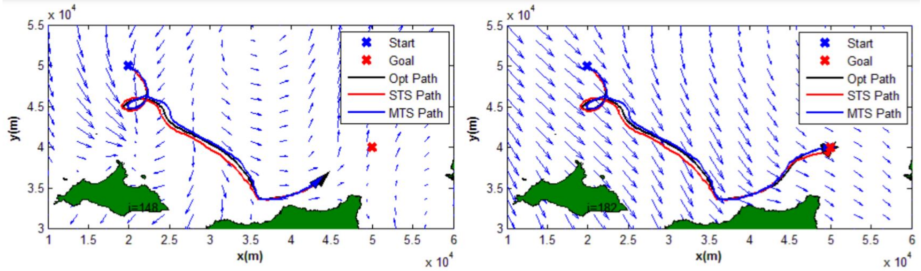


Figure 4: Image showing results of algorithm for time varying current generated from forecasts. Image: [9]

In both these articles the velocity of the vessel relative to the current was assumed to be very limited. This approach allows the optimisation algorithm to limit the amount of possible neighbours, meaning that the use of an A* algorithm does not take an unreasonable amount of time. This can be a good representation for vessels with limited controlability, but might not be suitable for vessels that are capable of moving independent of weather, as this would result in a computationally expensive graph search.

6.1.3 Seabed Coverage

A common task for autonomous vessels is to create bathymetric maps of the ocean floor. This is done by moving a sensor over all points in a desired area. This is a tedious, yet simple task, which makes it suitable for ASVs. In Glaceran et al., 2012, [10] a novel method for covering the area of interest was described. A common method used is the Morse-based cellular decomposition method. The working area is dividend into simply connected domains with a Morse based algorithm. In each simply connected domain a lawn-mower pattern path is then created to cover the entire area. A node network describing neighbouring domains are then created and a travelling salesman algorithm decides in which order the different domains are covered. An A* algorithm is made to find the shortest path between the end path of the previous domain and the start path of the next domain.

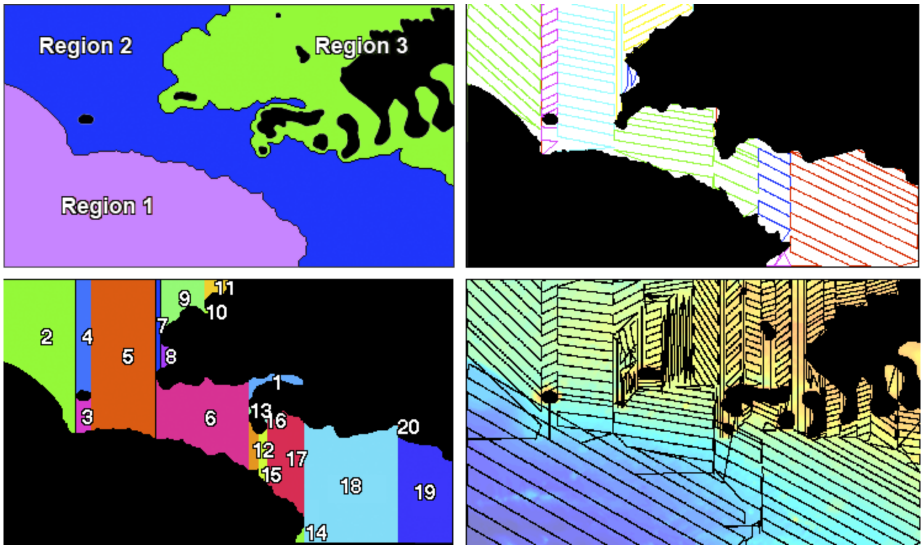


Figure 5: Seabed coverage method described in glaceran et al., 2012 [10]
1) Upper left: Working area divided into different regions of similar depth.
2) Lower left: region divided into smaller simple regions via the Moore method.
3) Upper right: Every simple region covered with a lawn-mower pattern.
4) Lower right: Entire region path connected together.

This simple method however does potentially create unnecessary overlap of sensor data. On most bathymetric sensors the field of view resembles a fan or cone. Area covered by the sensor will therefore change with distance between seabed and sensor. Seabed gradient also changes sensor coverage due to similar effects. To assure sufficient coverage, the planner has to create the lawn-mower pattern to fit the shallowest depth in the area. This however creates large overlap in deeper areas. To compensate for this a K means algorithm is used to divide the bathymetric map in k different areas of roughly similar depth. The created areas are then smoothed out by dilating and eroding the edges of the clusters. The Moore algorithm is then used for each for the clusters. To minimise change in elevation along the lawn mower patterns the paths are made perpendicular to the seabed gradient. Interlap spacing is also made to fit the shallowest part of the next sweep. A test comparing the two algorithms showed a reduction in path length form 15646.08 meters to 10349.63 meters. [10]

This example of adapting a mission to suit sensor use is a great example of how sensor data is implemented in path planning. This algorithm does however not take much into account vessel limitations, as the vessel expected to perform such a mission is expected to have close to full control of vessel position and velocity. This assumption is not always realistic for all types of vessels, as the controlability of self sufficient vessels is highly weather dependent.

6.1.4 Evolutionary Based Path Planning of an Autonomous Surface Vehicle

Using genetic algorithms to solve path planning problems have been successfully used in previous scenarios. In Arzamendia et.al. 2019 [11] the use of genetic algorithms for optimal area coverage was explored and compared to earlier studies and result.

In the example used, an autonomous vessel was to try to cover Ypacarai Lake by moving between beacons spread evenly around the lake edge. The Vessel coverage score was described by length of path L times a certain estimated width S. To prevent duplicate sampling of a same area, we subtract S^2 for every intersection of a previous path.

The vessel was constrained to follow either a Hamiltonian circuit, meaning that it can only go to each beacon once before returning to starting beacon, thus creating a travelling salesman problem. The other possible path was a Eulerian circuit, where the vessel is allowed to visit a beacon multiple times, but not the same path twice, as this would be a complete overlap.

Due to the high complexity of the optimisation problem, genetic algorithms have shown to be great at finding Quasi optimal solutions, meaning seemingly optimal solutions, but without any real proof of it being optimal. The first obstacle to overcome when using genetic algorithms is to start with a feasible population. Arzamendia uses an algorithm to find a feasible population which is then iterated until some condition is met.

The results showed that the genetic algorithm was capable of optimising the initial guessed paths. The Assumption of knowing all way-points of the vessel beforehand however is not applicable for the missions planned at NTNU, where offshore situations is of higher importance.

6.1.5 Artificial Potential Fields For Real Time Path Planning

In Yogang Singh et. al.(2017)[12] it was attempted to use potential fields to find a feasible path from a start to end position in an environment changing with time. A vector field is created by summing appropriate repelling and attracting vector fields into a complete vector field describing the work space. Repelling forces are placed at areas that the vessel has to stay away from. An attracting force is also place to incentivise the vector field to flow against the goal position. By summing the forces up you get a vector field, where the vessel can follow the field to ideally the desired point. Potential fields do not however guarantee a global minimum. Multiple geometries can create local minimums, making it hard for potential fields to work in the general case. The pro however is that it is not as computationally intensive as many other operations, which allows for more dynamic and reactive control. This also allows for a real time guidance algorithm.

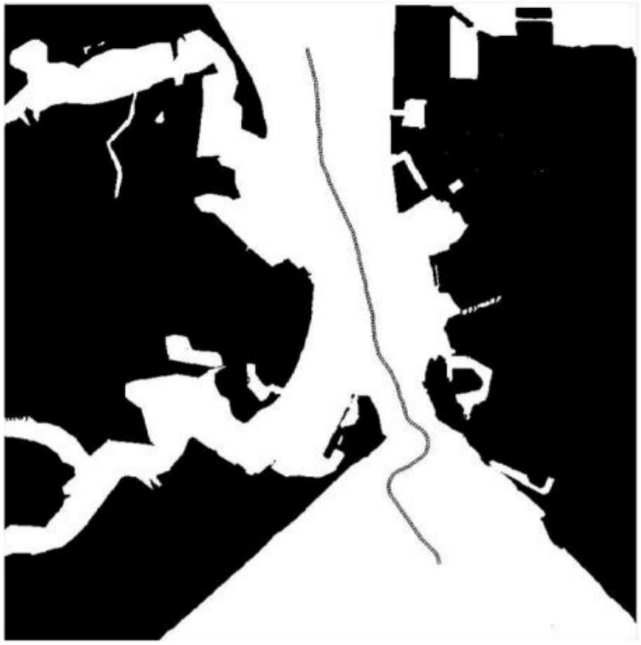


Figure 6: Potential field planning. Vessel starts at the bottom and moves up to the target at the top. Image: [12]

The algorithm was tested at the inlet of a lake. The algorithm was able to find its way through a narrow path without much issues. For simple cases tested in the article, a potential field is capable of finding an optimal path. However, the problem of local minima is not tackled, thus making it vulnerable to getting trapped.

6.2 A comparison of Optimization Techniques for AUV Path Planning

A lot of previous research as also been focused on path planning for autonomous underwater vessels. AUVs faces similar difficulties expected to be met by Surface vessels, as AUVs need to traverse large distances with a limited power reserve. In an article by Zeng et. al. 2016 [13], multiple different optimisation methods for AUVs were tested and compared. A 2D test arena was made to compare the different algorithms. There were two different arenas where one arena had obstacles while the other arena the obstacles removed. An identical vector field representing ocean current was added to both scenarios. An A* algorithm, RRT* algorithm and three different evolutionary algorithms, namely Particle swarm algorithm, quantum particle swarm algorithm, and a genetic algorithm were then tested on the two different scenarios.

The results shows that for both scenarios tested, the Quantum particle optimisation algorithms managed to find the most optimal solution. The other genetic algorithms also showed good results. The small differences in optimisation between genetic algorithms can be from both algorithm performance, but probably also down to implementation choices. The implementation of the A* only allows for the algorithm to move in 45 degree increments. The grid pattern the A* is allowed to move in does also seem to be low resolution. Therefore the different results between the different algorithms might not be as large as noted in the article, given a proper implementation of all algorithms. A more complex scenario would also be interesting to see, as the differences in algorithms become more visible in difficult scenarios.

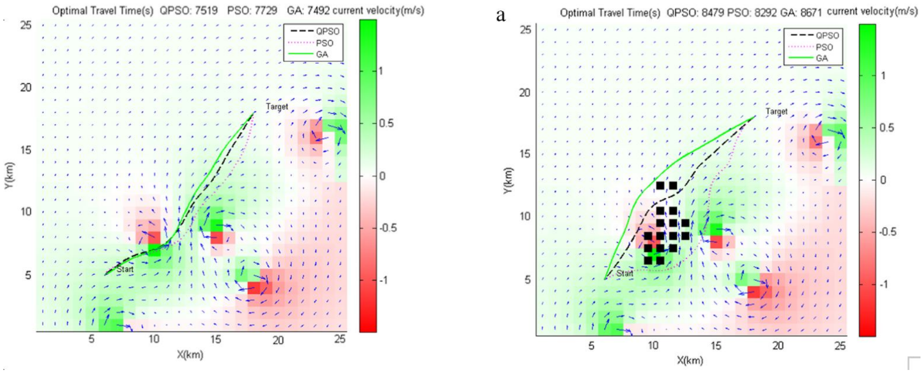


Figure 7: Results of path planning of genetic algorithms. Image: [13]

6.3 Neptus

The use of autonomous vessels has necessitated a software platform capable of supporting autonomous vessels in planning, control, information flow and post processing of accumulated data. As the autonomous fleets increases in complexity with heterogeneous vessels and and increased size of fleets, the demands for control software has increased. The Neptus framework is a framework made to be able to support this increasing complexity, while also supporting planning, operating, and the after action processes.

The requirements for Neptus is set up around the concept of operations (ConOps) requirement description method, which contains:

- Operational Setup

- Mission programming
- Mission execution
- Mission analysis

Operational setup, entails implementing a map of the Area f Operations and different constraints associated with the mission. Mission Programming contains programming of mission logic and plan to get the desired mission execution. Mission Execution: During a mission Neptus can be used to monitor the vessels an interact with the vessels while out on mission. Mission Analysis: After the mission Neptus can be used to analyse mission data.

Neptus can also be used to simulate missions, to that way be able to test out and preemptively iterate on the mission before real life testing[14]. Neptus is also the system used to control the AutoNaut and many other vessels used by NTNU.

6.4 Genetic Algorithms

6.4.1 Particle Swarm Algorithm

Due to the capabilities and adaptability of Particle swarm algorithms, this approach will also be the main approach for this master thesis.

Particle swarm optimisation is an evolutionary optimisation algorithm in which an optimal solution to a function is found by moving a set of particles around a parameter space. The position of the particles in the parameter space represent the parameters in an objective function. The parameters described by each particle gives each particle a score determined by the objective function. The particle are moved around in the parameter space in search of the parameters that gives an optimal score for the objective function. The movement speed and direction of the particles are given by the particles previous velocities and attraction forces that pulls the particles against the particles with the best local and/or global objective scores. [15]

Let $v = (v_1, ..., v_n)$ be a vector describing particle velocities, where each v_i is the velocity of particle i. The velocity and position of each particle is determined for each iteration by the equation:

$$
\begin{aligned}
v(n+1) &= \omega v(n) + \alpha(p_{best} - x(n)) + \beta(g_{best} - x(n)) && (6) \\
x(n+1) &= x(n) + v(n+1) && (7)
\end{aligned}
$$

Where $\omega,\ \alpha,\ \beta$ are continuous variables between 0 and 1. They can also be random variables from an even distribution from 0 to 1. The main issue is that the particles start to converge toward either a local or global optimal solution. This is a basic description of a particle swarm iteration. Many different equations and philosophies can be used to iterate the velocities. Particles swarm optimisation can also be used with mixed binary-continuous input parameters. This allows for binary operations to be implemented into the objective function. In binary particle swarms, binary operations such as AND $\cap$, OR $\cup$ and XOR $\oplus$ are used to iterate thorough different solutions. Similar to continuous particle swarm, the personal, local and global best solutions are used as attraction forces to make the solutions converge toward an optimal solutions. [16]

$$v(n+1) = \omega \cap v(n) \cup c_1 \cap (p_{best} \oplus x(n)) \cup c_2 \cap (g_{best} \oplus x(n)) \tag{8}$$

$$x(n+1) = x(n) \oplus v(n+1) \tag{9}$$

A full description of an implemented particle swarm algorithm can be found below. An actual implementation should however not be implemented directly as described. There is large computational gains possible from parallelising the algorithm.

Algorithm 1 Particle Swarm Optimisation

```
  n = numInputs
  m = numParticles
  t = maxIterations
Require: $x \in R^n X R^m$
  $x_{i,j}, i \in n, j \in m$
  $p_{best,p}, p \in m$
  $g_{best} \in R^1$
  for $k = 1 : t$ do
    for $j = 1 : m$ do
      for $i = 1 : n$ do
        $v_{i,j} = \omega v_{i,j} + \alpha(p_{best} - x_{i,j}) + \beta(g_{best} - x_{i,j})$
        $x_{i,j} = x_{i,j} + v_{i,j}$
      end for
    end for
    for $j = 1 : m$ do
      if $costFunction(x[:,j]) < p_{best,j}$ then
        $p_{best,j} = costFunction(x[:,j])$
        $x_{pBest,j} = x[:,j]$
      end if
    end for
    if $min(p_{best}) < g_{best}$ then
      $i = indexMin(p_{best})$
      $g_{best} = p_{best}[:,i]$
      $x_{gBest} = x_{pBest}[:,i]$
    end if
  end for
  return $g_{best}, x_{gBest}$
```

Particles swarm algorithms do not need to calculate derivatives. In optimisations algorithms such as SQP or newton method necessitates derivatives and double derivative, either by direct calculation or by estimation. Both are computationally expensive and not always feasible depending on the problem. Nonlinearities in a cost function may also throw off algorithms using derivatives to find a solution.

Part II
Theory for Vessel Modelling and Estimation

7 Modelling and Implementation of Wave Propulsion ASV

7.1 AutoNaut

The AutoNaut is an autonomous surface vessel that is made for long endurance missions up to two months at sea. The vessel is made by AutoNaut Ltd[17]. AutoNaut gathers energy during missions from solar panels placed on top of the vessel, and uses the wave induced change in attitude and height of the vessel to create propulsion. The product is a complete package including proprietary hardware and software. Scientific equipment, guidance and mission execution is all made by AutoNaut Ltd. The reasoning for NTNU to order the AutoNaut was the large payload available on the ASV compared to its size and weight. The AutoNaut 7 has the capacity of around 130 Kg of payload for 250 kg displacement, which gives a 0.52 payload to weight ratio compared to a competitor like Saildrone at around 0.2 payload to weight ratio[6].

NTNU wanted to build a open architecture research vessel to base their research for both oceanic and cybernetic research to be published. Therefore the AutoNaut was gutted of its internal hardware, and replaced with an open source architecture carefully described in multiple articles and the AutoNaut Wiki [18]. To get a good introduction to the NTNU version of the autonaut, a transcript form Dallolio et al.,2019 [3] page 3-8.

7.2 *Autonaut Description*

In order to accomplish missions with the profiles described, the AutoNaut is equipped with a scientific payload that targets the environmental parameters of interest. The vehicle is provided with a propulsion system that entirely relies on sea surface waves [1]. Two pairs of spring-loaded submerged hydrofoils are connected at the bow and stern by two vertical struts. When a surface wave lifts the bow or the stern of the vehicle, the corresponding strut lifts the foils, which are subsequently pulled back by the spring generating a forward thrust. This self-propelling mechanism limits the speed achieved by the vessel during operations up to 3-4 knots. However, the platform is equipped with a small thruster that can be actuated by the collision avoidance algorithm to enable sharper manoeuvres or whenever surface waves are too small to produce acceptable propulsion. The heading of the vessel is controlled by means of a rudder commanded by the navigation control unit, and can turn up to 45 degree relative to its centred position. The hull is divided into two main water-tight compartments, where batteries, computers and some sensors are hosted. However, most of the sensors needed for navigation and environmental data collection are placed outside the compartments 9. The scientific payload is described in Table I. Except for the Weather Station (Airmar 120WX) which is connected to the vehicle mast, all other sensors are placed on the submerged keel ...

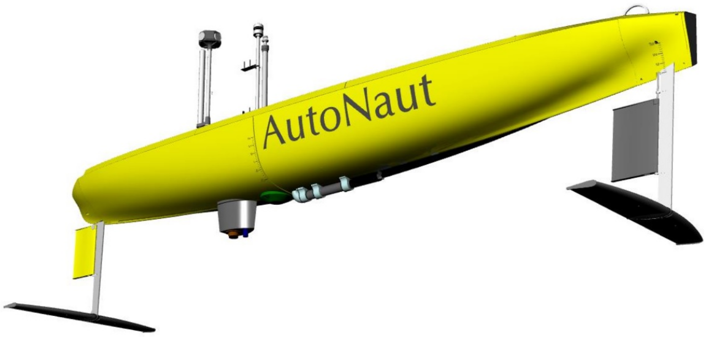


Figure 8: A CAD model of the AutoNaut

7.2.1 *Energy storage and distribution*

The upper surface of the hull is covered with three Solbian SP 104 solar panels, whose maximum output power rating is 104W each. The onboard battery bank is made of four 12V 70Ah Lead Gel batteries, wired in parallel as most of the components require around 12V. In order to control the power produced by the panels, two Maximum Power Point Tracking (MPPT) controllers are chosen. These have built-in inverters and can step the voltage up or down prior to supplying the batteries. This is required as the solar panel output varies with the observed load impedance. Two step-down MPPT controllers is used in the power system. Panel 3, which is furthest from the mast, is connected to one controller because it is unlikely that the internal bypass diodes are activated due to shading, meaning that the panel output always will be higher than the required input voltage for the controller. The panels near the mast which are likely subject to partial shading, are connected in series to another step-down MPPT controller. The chargers input will thus always be higher than the minimum voltage requirement, even if both arrays in one panel are bypassed. The units selected are Victron BlueSolar MPPT 75/15. Fig. 5 provides an overview of the structural design of the power management system implemented into Level 1 unit housing. An external toggle switch allows to disconnect the load power line that provides power to all components. This means that when a mission is completed and the user turns off the computers and sensors, the batteries can still be recharged by the solar panels through the controllers. Fig. 5 also shows *how the power is distributed to the whole system. The CR6 Campbell Scientific Datalogger, compass GPS, Iridium and Rudder Servo are directly connected to the load port of BlueSolar 1, through the switch. However, they are controlled by the CR6. Level 2, Level 3, AIS transceiver, 4G/LTE Modem, SentiBoard timing unit, Radar Reflector and Pumps are instead powered through solid state relays that are digitally controlled by CR6 GPIOs. The OWL VHF radio is the only component being directly powered by a 12V output port of the CR6. Historic data for solar radiation during fall in Trondheim ...*

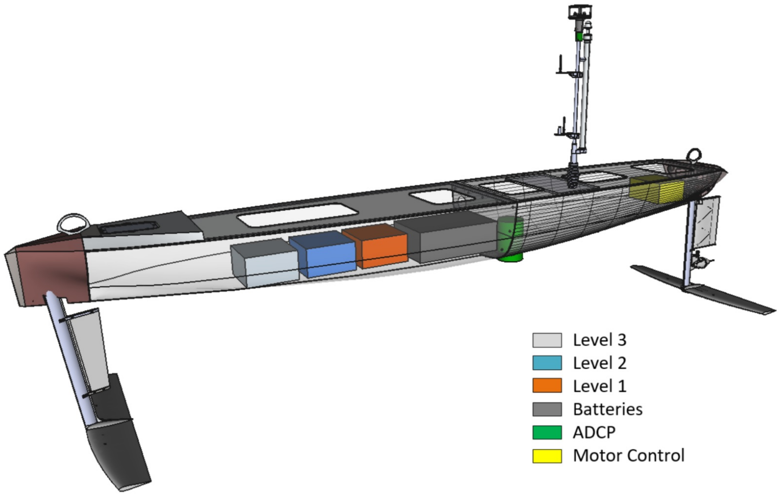


Figure 9: The different compartments and hardware in the AutoNaut

7.2.2 Communication

B. Iridium Communication: The vessel is equipped with two separate Iridium Rock- block+ units that host an Iridium 9602 transceiver, an antenna and a voltage regulator. As shown in Fig. 4, both Level 1 and Level 2 can send a receive messages over satellite. This communication link supports the mission when 4G/LTE coverage is absent and involves less mission flexibility and higher costs. Level 1 periodically sends a message reporting the overall state of the system: Time and location, power settings, battery voltage, consumed and produced power. The operator is therefore able to communicate changes in the power settings of the vehicle and restart sensors and components. The Rockblock+ unit connected to Level 2 is instead used to communicate new or modified plans to the onboard software (Dune). The vehicle acknowledges the reception of the plan and later its outcome. This solution has a limited bandwidth and is therefore only suitable for simple control monitoring or tracking applications. The maximum package sizes are 340 bytes for sending and 270 bytes for receiving. Although *the latency is typically a few seconds, it may increase to up to a minute or more depending on the remoteness of the area and the available satellites.*

C. VHF Radio Communication Onboard the vehicle, an OWL VHF radio transceiver allows efficient point-to-point communication between the operators and Level 1. It supports a large variety of modulation types and encoding, that can be configured through a serial port. A Java GUI (Fig. 10) enables manual control and direct monitoring of the vehicle, over VHF. During a mission, this link is turned off in order to save energy. It is however turned on when manual control of the vehicle is needed. An automatic routine enables the radio whenever a fault is detected. The radio transmits the location and power settings, allowing the operators to find the vehicle and manually control it to shore. A passive duplexer allows the OWL VHF radio to share one antenna with the AIS. Unlike an active splitter, the duplexer has a notch filter in each port that attenuates the frequency used by the other port. This means that both radios can always transmit without hearing each other and everything is sent out on the antenna. The filters are tuned to specific frequencies, so the radios cannot change frequency. The selected cut- off frequency of the AIS port is 162MHz (center of AIS frequencies 161,975MHz and 162,025MHz) ...

8 Vessel Dynamics Model

8.1 Assumptions

In Fossen et.al. [19] A full rigid body dynamics model for ships include surge, sway and heave, as well as pitch, yaw and roll. Pitch, roll and heave are self stabilising under normal conditions, and therefore often set to 0. This simplification comes at the expense of removing the dynamics AutoNaut uses for propulsion. The propulsion dynamics are instead modelled as a force independent of the vessel modelling. By setting Pitch, roll and heave to 0, dynamics simplifies down to surge, sway and yaw. Yaw is controlled by a rudder on board the AutoNaut. Given that the rudder is capable of accurately controlling yaw, we can expect that heading equals desired heading within the expected time step of the simulations, which are expected to range between minutes to hours. Yaw dynamics are therefore neglected. Movement on the surge sway plane can be described by Newtons second law. If it is assumed that the only forces acting on the vessel is current, wind and forces form the AutoNaut hydrofoils, the equation becomes:

$$M\dot{\vec{v}} = \vec{F}_{waves} + \vec{F}_{wind} + \vec{F}_{currents} \tag{10}$$

Wind and current forces can be described by the drag equation. From Fossen et.al. [19] a drag equation is explained as the equation $-sgn(V)\frac{1}{2}\rho C_d A V^2$. ρ $\frac{kg}{m^3}$ describes the density of the medium being passed through. C_d is the adjustable variable fitting the measured drag forces to the model. A m^2 describes the cross-section of the object as projected onto a plane orthogonal to the force direction.C_dA changes depending on what angle the vessel passes through the medium. As the vessel is moving at velocities between 0 - 1 m/s it can be assumed that the flow passing the vessel will be close to laminar. The operational range of the velocity is also relatively small, therefore a linear drag model is assumed to be a sufficient assumption. This will also simplify parameter estimation and the ease computation.

8.2 Model Fitting

Making an accurate model for the vessel allows the model to predict vessel behaviour in different conditions. This predictability enables an algorithm to calculate the duration of a path given weather conditions either measured or gathered from weather forecasts. It will also be able to predict unfeasible paths, which is a vital part of path planning. The Linear model can be described by:

$$m\vec{a} = D_w(\vec{V}_w^n - \vec{V}^n) + D_c(\vec{V}_c^n - \vec{V}^n) + F_{waves}$$

The drag coefficients D_w and D_c is expected to depend on the angle between vessel and fluid. The drag force is expected to be largest when orthogonal to the vessel and smallest along the surge direction. To recreate this property the relative current and relative wind will be decomposed to two velocities to relative velocity parallel and orthogonal to the bow-stern line. This coordinate system is normally described as the body coordinate system. Given relative velocity in body coordinates drag forces can be set as two parameters. The drag forces can then be described in body coordinates as:

$$F_d^b = \begin{bmatrix} D_x & 0 \\ 0 & D_y \end{bmatrix} \begin{bmatrix} V_{rx}^b \\ V_{ry}^b \end{bmatrix} \tag{11}$$

The relative velocity has to be changed to a description of Earth fixed Vessel and weather velocities to be able to us forecast and vessel data:

$$\vec{V}_r^b = (\vec{V}_w^b - \vec{V}^b) \tag{12}$$

If we want to describe dynamics In NED earth fixed coordinates, the equation changes to:

$$F_d^n = R_b^n \begin{bmatrix} D_x & 0 \\ 0 & D_y \end{bmatrix} R_n^b \begin{bmatrix} V_{wx}^n - V_x^n \\ V_{wy}^n - V_y^n \end{bmatrix} \tag{13}$$

This can be implemented for both wind an current forces. By inserting drag and wave forces into equation, we get:

$$m\vec{a}_n = R_b^n D_w R_n^b(\vec{V}_w^n - \vec{V}^n) + R_b^n D_c R_n^b(\vec{V}_c^n - \vec{V}^n) + R_b^n \begin{bmatrix} F_f \\ 0 \end{bmatrix} \tag{14}$$

$$mR_n^b\vec{a}_n = D_w R_n^b(\vec{V}_w^n - \vec{V}^n) + D_c R_n^b(\vec{V}_c^n - \vec{V}^n) + \begin{bmatrix} F_f \\ 0 \end{bmatrix} \tag{15}$$

Equation 15 can be further used to estimate the parameters for the AutoNaut. Current model is dependent on vessel heading for all estimated parameter. Using a model explicitly dependent on vessel heading creates an unwanted complexity. If the vessel wants to follow a predefined course, the vessel need to find a correct creep angle. To avoid the need of directly calculating creep angle, a simpler model is needed. As previously commented the timesteps are expected to be between minutes to hours. The vessel is therefore expected to reach a steady state velocity, therefore setting acceleration to 0. Given that the drag forces from wind are low compared to current forces, the force generated from the ocean will be perpendicular to the forces generated by the hydrofoils. in this case the only relevant drag parameter will be the forward drag of the vessel, as there will be no sideways movement relative to the current. It is still however important to estimate sideways and forward drag separately, otherwise a parameter estimation algorithm will estimate the drag as a sum of forward and sideways drag. If the difference between sideways and forward drag is large, the error in the parameter estimate will also be large, and highly dependent at which angle the relative current will be during parameter estimation. Replacing the heading dependent drag coefficients with with D_w, D_c and setting acceleration to 0, we get:

$$0 = D_w(\vec{V}_w^n - \vec{V}^n) + D_c(\vec{V}_c^n - \vec{V}^n) + F_{wave} \tag{16}$$

$$(D_w + D_c)\vec{V}^n = D_w\vec{V}_w^n + D_c\vec{V}_c^n + F_{waves} \tag{17}$$

$$\vec{V}^n = \frac{1}{(D_w + D_c)}(D_w\vec{V}_w^n + D_c\vec{V}_c^n + F_{waves}) \tag{18}$$

$$\vec{V}^n = \frac{1}{(D_w + D_c)}(D_w\vec{V}_w^n + D_c\vec{V}_c^n) + \frac{1}{(D_w + D_c)}F_{waves} \tag{19}$$

Thus, the resulting earth fixed NED velocity can be described as the sum of the velocity gained from drag forces and the hydrofoil forces. To simplify notation the velocity generated from weather will be called disturbance velocity, and velocity gained from hydrofoil forces will be called wave velocity.

8.3 Wave Propulsion Modelling

The AutoNaut uses a novel propulsion method, where the roll and pitch of the vessel is converted into forward propulsion. Equipping the AutoNaut with sufficient solar panels for on board components gives the vessel almost unlimited operational range. Fouling and components wearing out do however limit the practical limits of the missions to a couple of months[17]. The wave propulsion hardware consist of two vertical rods connected to the bow and stern of the vessel. There are one hydrofoil on the port and starboard at the bottom of each rod. These hydrofoils are connected to the rod via a rotating shaft with a spring forcing the foils to the neutral position. The centre of pressure on the hydrofoils are slightly aft of the rotation centre-line, thus when the AutoNaut rolls or pitches, the pressure from the surrounding water will offset the hydrofoils form the neutral position. In this offset position the pressure forces point forward at an angle. Gravity and buoyancy neutralisers all vertical forces, while the horizontal forces in the surge direction pushes the AutoNaut forward. The hydrofoils surface is orthogonal to the sway axis, thus there are no forces in the sway axis. Vessel symmetry around the Surge, Heave plane also nullifies any sideways forces. Propulsion forces in the body frame therefore simplifies down to:

$$F^b_{wav} = \begin{bmatrix} F_{wav} \\ 0 \end{bmatrix} \tag{20}$$

The hydrofoils on the AutoNaut only create forces when the waves induce a roll, pitch or heave motion on the vessel. Given that wave velocity and curvature is highly dependent on wave frequency, thus it can be assumed that the energy transferred from the wave to the AutoNaut can be described by a transfer function.

The velocity generated from the hydrofoils is assumed to be largest when the heading is parallel with wave direction. The smallest velocity is generated when the waves are orthogonal to the heading. When the angle is between these positions the velocity is somewhere between forward and side ward force as well. The generated wave velocity is therefore expected to represent an oval if plotted in a polar diagram. Given that the polar diagram plots for relative angle between vessel heading and wave direction, the polar plot will be represented by an oval with largest radius along the 0 and 180 degree angle. The smallest radius will be along the 90 and 270 degree angle. The resulting earth fixed velocity will then be represented by the sum of the disturbance and wave velocity vector, as represented in figure 10.

The coordinate system used in figure 10 is a plane coordinate system along the water surface. X axis being parallel to the wave direction. The motivation behind rotating the coordinate system along the wave direction is to simplify equations that calculates the given velocity for a certain course. The main simplification being the equation representing the hydrofoil force can be described by an oval with c radius along the x axis and d radius along the y axis:

$$1 = \frac{x^2}{c^2} + \frac{y^2}{d^2} \tag{21}$$

If you want to to centre the oval on (x0,y0) you can replace x and y with $(x - x_0)$ and $(y - y_0)$. Let V_{wave} be described by an oval with radii V_f and V_s.

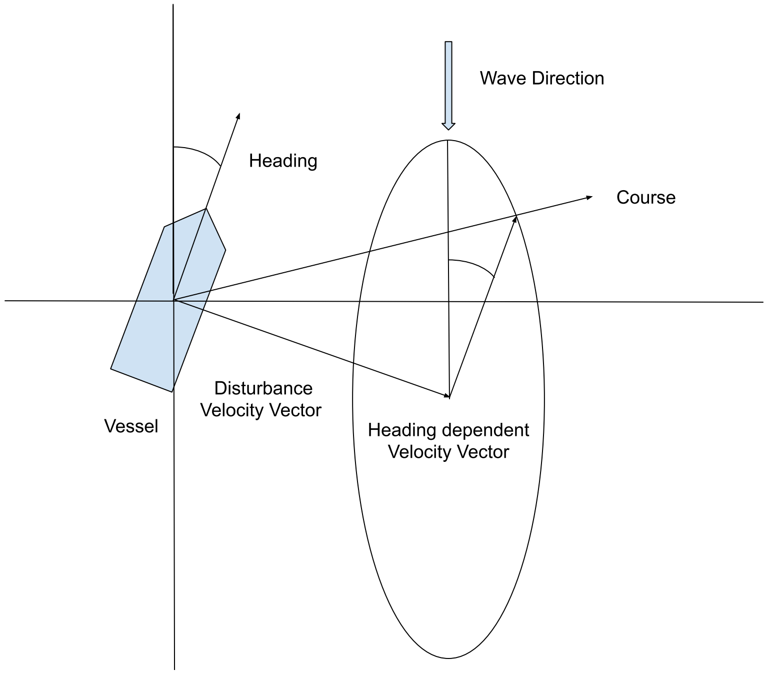

Wave Direction
Heading
Course
Vessel
Disturbance Velocity Vector
Heading dependent Velocity Vector

Figure 10: Figure describing the correlation between heading, course and velocity.

Since the radii should change with the amplitude and frequency ω of the waves, we multiply the radii with the function K_{waves}. Thus V_{wave} can be described as the vector from the origin to the perimeter of the oval:

$$1 = \frac{(x - v^w_{x,ned})^2}{c^2} + \frac{(y - v^w_{y,ned})^2}{d^2} \tag{22}$$

$$c = K_f(\omega)\frac{1}{(D_w + D_c)} \tag{23}$$

$$d = K_s(\omega)\frac{1}{(D_w + D_c)} \tag{24}$$

$$\tag{25}$$

The energy harvested form the hydrofoils to only push along the surge body axis, as previously discussed. An exact model for hydrofoil forces are however harder to come by. Rolling to hydrofoil forces have not been in much focus. To get an understanding of wave propulsion, a short introduction to wave energy is needed.

In linear wave theory a one dimensional wave is described by the equation:

$$\eta(t, x) = a\cos(\omega t - kx) \tag{26}$$

Where x defines the position along the one dimensional wave, t defines the time, $\eta(t, x)$ defines the height of the surface of the wave. ω and k depends on the depth of the ocean. Assuming that we have a irrotational incompressible fluid, it is a descent assumption that the waves can be described by a potential field. Given the velocity potential field $\Phi(x, z, t)$ velocity in x and z direction should be given by the directional derivative of the Velocity potential field, thus

$$\frac{\partial \Phi}{\partial x} = u_x, \quad \frac{\partial \Phi}{\partial z} = u_z \tag{27}$$

The rest of the constraints are linearised and and it is assumed that the wave amplitude is small compared to the wavelength of the waves. Given these assumptions the constraints can later be solved.

Due to the fact that water is an incompressible fluid, the divergence of the potential field will equal zero at all points, thus

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial z^2} = 0 \tag{28}$$

The bottom of the ocean is assumed to be impermeable, thus any vertical motion at the sea bottom is impossible, giving us the constraint

$$\frac{\partial \Phi}{\partial z} = 0, \ at\ z = -h \tag{29}$$

Given the scale of the ocean, it is assumed that waves height to ocean depth is infinitesimal, thus the vertical motion of the wave surface equals the vertical flow velocity.

$$\frac{\partial \eta}{\partial t} = \frac{\partial \Phi}{\partial z},\ at\ z = \eta(t, x) \tag{30}$$

The air pressure over the wave surface is assumed to be constant. Via the unsteady Bernoulli's equation you get the linearised final constraint for small waves:

$$\frac{\partial \Phi}{\partial t} + g\eta = 0 \tag{31}$$

A general solution for these constraints has until now been impossible to solve, but for more specific situations a solution has been found. Given our description of a wave $\eta(x, t)$ a solution for Φ can be described by

$$\Phi = \frac{\omega}{k}\, a\, \frac{\cosh\big(k\,(z+h)\big)}{\sinh\,(k\,h)}\ \sin\,(kx\ -\ \omega t), \tag{32}$$

$$\omega^2 = g\,k\ \tanh\,(kh) \tag{33}$$

From the constraints one can observe that wave frequency and velocity depend on eachother. Given that depth of the ocean is much larger than the wavelength $(1 << kh)$ we get

$$\omega^2 = gk \tag{34}$$

Thus the wave can be described by only wave height and frequency. For most wave forecasts you normally only get wave height and frequency, which given smaller waves is enough to describe the whole wave system.

The energy contained in a wave can be roughly described as:

$$E_w = \frac{1}{2}\rho g h^2 \tag{35}$$

The amount of energy extracted from the waves via hydrofoils has from tests not been observed to increase quadratic with wave height. The limited length of the Hydrofoil rod also limits the achievable energy gained from the increased height of waves. A linear correlation between wave height and hydrofoil force will therefore be a placeholder until a better understanding of the hydrofoils can be made. The resulting wave velocity model will be described as:

$$1 = \frac{(x - v^w_{x,ned})^2}{c^2} + \frac{(y - v^w_{y,ned})^2}{d^2} \tag{36}$$

$$c = K_f \frac{h}{(D_w + D_c)} \tag{37}$$

$$d = K_s \frac{h}{(D_w + D_c)} \tag{38}$$

$$\tag{39}$$

8.4 Implementation of Parameter Estimation

Equation 15 has six unknown parameters. D_w, D_c, $K_f(\omega)$ and $K_s(\omega)$, as D_w and D_c are diagonal 2x2 matrices. This system is a linear time varying model, thus a linear least squares estimate can be used to estimate the parameters, given enough equations. The AutoNaut tracks velocity for every second, thus a large representative data set is therefore available. To estimate the unknowns, we want to set the unknowns as x in an Ax=b equation. If we want to to write this as a linear equation we set A, x and b as such:

$$V^b_{wi} = \mathbf{diag}\left(R_z(\psi_i)(\vec{V}^n_{wi} - \vec{V}^n) \right) \tag{40}$$

$$V^b_{ci} = \mathbf{diag}\left(R_z(\psi_i)(\vec{V}^n_{ci} - \vec{V}^n) \right) \tag{41}$$

The hydrofoil forces are estimated to be the radius of an ellipse for a certain angle. This is however hard to the directly estimate, as the radius of an ellipse in polar coordinates can be described as:

$$r = \frac{ab}{\sqrt{(a\cos(\theta))^2 + (b\sin(\theta))^2}} \tag{42}$$

where a and b is maximum and minimum radius of the ellipse and θ is the polar angle. It is hard to explicitly estimate a and b, thus there is easier to use a similar form to estimate the force. The discrepancy from using a slightly dissimilar model to an oval, is expected to be trivial due to the ellipse already being an estimate created from deductions during ocean tests. To simplify manners the ellipse will be estimated by a ellipse lookalike form. A slightly similar shape to an ellipse is the model:

$$r = a\cos(\eta_i)^2 + b\sin(\eta_i)^2 \tag{43}$$

As discussed in the linear wave theory chapter, the forces created from a wave is estimated to be modelled as the the wave height multiplied by a constant.

The frequency of the wave is also expected to affect the vessel velocity, but to keep the model simple, it is assumed to be frequency invariant. If the forces are not frequency invariant, the estimated forces should differ for different wave frequencies, so it should be possible to spot during parameter estimation. The wave forces were therefore implemented as follows:

$$r_i = \begin{bmatrix} \cos(\eta_i)^2 W(t)^2 & \sin(\eta_i)^2 W(t)^2 \\ 0 & 0 \end{bmatrix} \tag{44}$$

Then A, x and b can be described as :

$$A_i = \begin{bmatrix} V^b_{wi} & V^b_{ci} & r_i \end{bmatrix} \tag{45}$$

$$x = \begin{bmatrix} D_{wf} \\ D_{ws} \\ D_{cf} \\ D_{cs} \\ K_f(\omega) \\ K_s(\omega) \end{bmatrix} \tag{46}$$

$$b_i = mR_z(\psi_i)(\vec{V}^n_{i+1} - \vec{V}^n_i) \tag{47}$$

Since we want a least square estimate we want to have as many equations as possible. For the least squares to be able estimate the parameters, the system will have to be observable, meaning that the system should only have one unique solution. For this to be correct, A has to be full rank, and b to be nonzero. Thus, at least 3 unique time steps to be able to find a solution, but the more the better. The resulting equation can be set up as:

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix} \tag{48}$$

$$x = x \tag{49}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \tag{50}$$

$$\tag{51}$$

Due to the rotation matrices and uncertainties in both measurements and forecasts, the result does not necessarily converge to the true parameters. When testing out the least squares solution, on simulated analytical models tests showed that the method was hard pressed to find correct parameters when noise was included into the measurements. Especially when included to the GPS data. For those interested, the estimation simulation is added with the code attached to the thesis.

8.4.1 Implementation

During mission the AutoNaut logs data gathered from multiple different instruments. To estimate vessel parameters, multiple of these sensors can be used to aid in the modelling. The basic data needed for making a parameter estimation is vessel velocity and vessel heading. Current, wind and waves can be gathered from online weather forecasts. The AutoNaut logs its GPS position up to two times every second, and therefore logs a large amount of GPS data. The vessel heading however is not sampled at similar rates. Values therefore have to be interpolated to match time wise. Sample time is however so quick that neither vessel velocity or heading is expected to change much between time steps given that every time step is used. Therefore interpolating data is expected to give an accurate representation as well.

Vessel velocity can be calculated by dividing change in GPS position for each time step. Tests on analytical models shows large discrepancies in parameter estimation if velocity estimates are off, therefore this should be alleviated in the test data. As noise on GPS position is expected to be constant, relative noise can be reduced by increasing the distance/time step between every sample. Since we are dependent on change in velocity to estimate, this should be able to improve the estimates.

When using real sampled data, the data set often have some corrupted data cells. In the case of corrupted data for a given time-step, all data for that time step was removed. Since every recorded data has a timestamp, it is still easily possible to use the data even with larger gaps in the data.

Algorithm 2 Parameter Estimation

```
$\psi$ = loadHeading()
$p_n$ = loadGpsPositions()
$v_n$ = estimatedVelocity(position)
$a_n$ = estimatedAcceleration(position)
$Weather$ = loadWeather(position)
for i = 1:n do
    $V^b_{ci} = diag(Rotation(\psi[i]) * (Weather.current[i] - v_n[i]))$
    $V^b_{wi} = diag(Rotation(\psi[i]) * (Weather.windAngle[i] - v_n[i]))$
    $\eta = Weather.waveDir - \psi[i]$
    $r_i = \begin{bmatrix} \cos(\eta_i)^2 W(t) & \sin(\eta_i)^2 W(t) \\ 0 & 0 \end{bmatrix}$
    $A[i] = \begin{bmatrix} V^b_{wi} & V^b_{ci} & r_i \end{bmatrix}$
    $b[i] = mass * Rotation(\psi_i) * (\vec{V}^n_{i+1} \vec{V}^n_i)$
end for
$parameters = b * A^{-1}$
return parameters
```

8.4.2 Estimating parameters with vessel sensors

The AutoNaut is equipped with a Doppler current sensor, wind sensor and and IMU, that should give the AutoNaut the possibility of estimating all needed measurements to estimate the parameters[3]. Wave estimation is still not finished on the AutoNaut, but relative wind and relative current can already be measured by the wind and current sensor. Therefore instead of using weather forecasts for wind and current it can be calculated relative to vessel body. These measurements will be in body frame, thus the equations changes slightly to:

$$V_{wi}^{b} = \mathbf{diag}\Bigg(V_{wmi}\Bigg) V_{ci}^{b} = \mathbf{diag}\Bigg(V_{cmi}\Bigg) \tag{52}$$

where V_{wmi} is measured wind relative to vessel, and V_{cmi} is measured current relative to vessel. If the estimates are good, one should expect measurements to be similar in both cases, so it will be interesting to see. If they are different, it does not discredit any of the methods, but but it is likely that one of them are erroneous.

Part III

Optimal Mission Planner Implementation

9 Velocity Model Use in Optimisation Algorithms

When optimising for an optimal path, multiple approaches can be made. Different approaches have different pros and cons. Adjustable parameters for optimisation could be vessel heading, rudder angle, vessel course or way-points. There are however definitive properties that makes way-points the best solution.

Parameters using vessel dependent input-parameters is an intuitive starting assumption. Normal control inputs for a seagoing vessel is often rudder angle, heading, or course. However, vessel dependent inputs are not able to predict vessel positions, as the vessel position will depend both on vessel inputs, as well as unknown parameters such as weather and duration of time step. Changes in weather or time steps can change resulting vessel position, thus finding a parameter input that takes the vessel to a predefined destination, is a complex task.

To alleviate this problem way-points are used as parameter input. Using way-points to determine vessel course means that the path for the vessel is predetermined explicitly. A drawback for this type of path description is that not all parameter inputs are feasible. If weather prevents the vessel from being able to follow a path between two way-points, the estimated duration between points will not be solvable. This infeasibility does however represent reality, as an infeasible solution represents an impossible path.

For optimisation algorithms using derivatives to find feasible solutions, discontinuitites in the objective function will lead to problems. For evolutionary algorithms however this is a desirable property. Infeasible solutions are easily discarded and replaced with other feasible solutions. The large nonlinear chaotic properties of weather [20, 21] and cost function makes derivatives complex and prone to errors. Thus genetic algorithms are already the desired optimisation algorithm for the problem.

9.1 Bearing, Courses and the Great Circle

When planning paths traversing large distances, the curvature of the earth begins to come into effect. AutoNaut is expected to travel for long distances covering large areas of the Norwegian coast. A pane linearisation of the ocean surface begins to become inaccurate. One degree change in longitude at 60 degree latitude equals around 56 kilometres. At 70 degrees latitude however one degree in longitude equates to 38 kilometres. This keeps decreasing until you get at 90 degree latitude where one degree change in longitude equals 0 meters. There are multiple ways of taking into account the curvature of the earth. One solution is to linearise for each way-point in the path. This will however introduce changing errors depending on the distance and on angle between way-points. Another solution is to model vessel position by Quaternions, or using rotational velocity to estimate and simulate position. However, since the raw data generated in the vessel sampling is in longitude and latitude, the algorithms will have to include calculation with latitude and longitude. way-points in longitude and latitude are also a user friendly format for operators to use and is readily available on most maps, including google maps and similar services. For these reasons the great circle calculation method was used.

In great circle calculations, one take advantage of the property that the shortest path between two way-points is equal to the path described by a circle with origin at the earth centre intersecting the two way-points. Therefore the shortest path between two way-points can be described accurately by an analytical function. If a vessel follows the shortest distance path between two points, the bearing will not stay constant either. This is also able to accurately calculate using great circle mathematics, simplifying overlaying of weather forecasts to position depending on vessel heading. The method is also shown to be accurate even at small changes in latitude and longitude. [22, 23]

To calculate the distance between two way-points (φ_1, λ_1) and (φ_2, λ_2), where φ_i is latitude, λ_i is longitude, and r is radius of the earth, the distance will be:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{53}$$

To calculate the bearing at the starting way-point, also called the forward Azimuth, the angle can be calculated as:

$$\theta = \text{atan2}(\sin(\lambda_2 - \lambda_1)\cos(\varphi_2),\ \cos(\varphi_1)\sin(\varphi_2) - \sin(\varphi_1)\cos(\varphi_2)\cos(\lambda_2 - \lambda_1)) \tag{54}$$

These calculations are more computationally heavy than many other methods. There are however not currently any external factors setting limits for computational time, and a large part of the computational is is focused around loading and slicing large weather forecast maps spanning the entire Norwegian mainland coast. Future tests may look further into ways of simplifying these calculations, but at the moment, accuracy was deemed more important.

9.2 Finding Path Duration for a way-point Path

Given two way-points P_1 and P_2, an estimate for the time needed for the vessel to travel from P_1 to P_2 has to be found. First step calculate the course that takes the vessel from P_1 to P_2. Since the earth is a sphere, longitude and latitude is used to describe position. The great circle equations is then used to calculate course and distance between two way-points. When the desired course is found, the velocity for the course is calculated. For the vessel to move straight between the two way-points, the vessel velocity vector has to be parallel to the desired course. Given the velocity model explained in figure 10, it can be observed that velocity vectors parallel to the desired course will be described by the intersection between the course line, and the wave velocity oval. Intuitively one might also see that not all course will give an intersection between the course line and the oval. In these cases, it will not be possible for the vessel to travel straight between the two different way-points. However, given that there is an intersection between the line and oval, the intersection can be described by the following equations. $1 = \frac{(x-x_0)^2}{c^2} + \frac{(y-y_0)^2}{d^2}$ describes the oval, and y = ax + b describes the desired course line. a can be found by: $a = \tan(\theta)$, where theta is the desired course in the wave coordinate system. in the case where a goes to infinity the solution can be found by setting x=0. By inserting ax for y in the oval equation we get a second order polynomial equation describing the intersection points:

$$1 = \frac{(x-x_0)^2}{c^2} + \frac{(y-y_0)^2}{d^2},\ y = ax + b \tag{55}$$

$$1 = \frac{(x-x_0)^2}{c^2} + \frac{(ax+b-y_0)^2}{d^2} \tag{56}$$

$$1 = \frac{x^2 - 2xx_0 + x_0^2}{c^2} + \frac{(ax+b)^2 - 2(ax+b)y_0 + y_0^2}{d^2} \tag{57}$$

$$1 = \frac{x^2 - 2xx_0 + x_0^2}{c^2} + \frac{a^2x^2 + 2abx + b^2 - 2axy_0 - 2by_0 + y_0^2}{d^2} \tag{58}$$

$$0 = x^2\left(\frac{1}{c^2} + \frac{a^2}{d^2}\right) + x\left(-\frac{2x_0}{c^2} + \frac{2ab - 2ay_0}{d^2}\right) + \frac{x_0^2}{c^2} + \frac{b^2 - 2by_0 + y_0^2}{d^2} - 1 \tag{59}$$

The equation will give us two solutions for x. If there are complex solutions, the course is considered infeasible. Given that there are two solutions, the resulting velocity vector can be calculated by:

$$\vec{V}_w^n = \begin{bmatrix} x \\ ax \end{bmatrix}$$

The a can be found by using the equation To get the velocity along the course, we can multiply the velocity vector with a course unit vector:

$$V = \vec{V}_w^{n\top} \begin{bmatrix} \cos(\psi_w^n) \\ \sin(\psi_w^n) \end{bmatrix}$$

If both speeds are negative, it is considered to be infeasible. If there are one or more positive speeds, the fastest speed will be used. Given the special case where there is only one solution, the algorithm will handle the solution as two similar solutions, and work as normal.

To find the duration needed to travel from P_1 to P_2 we now calculate the distance by using the great circle distance equation. The duration is calculated by dividing distance on the velocity. For a path with multiple way-points, the procedure is done successively for each pair of way-points describing the path.

This can not be done in parallel, since the velocity is weather dependent, and therefore also time dependent. Thus all previous way-points have to be calculated before the correct velocity can be calculated. Total duration can then be found by summing the duration between each way-point. If any of the steps are unfeasible the whole path will become unfeasible. Thus when calculating the duration of a path, it can be discarded if any part of the path is unfeasible. This way of calculating Earth fixed velocity is similar to equations described in [8], but modified to fit with wave propulsion modelling.

Algorithm 3 Vessel Velocity

```
DisturbanceVelocityNed = (Dc*current + Dw*wind)/(Dc + Dw)
DisturbanceVelocityWave = Rotation(waveAngle)*DisturbanceVelocityNed
x0 = DisturbanceVelocityWave(1)
y0 = DisturbanceVelocityWave(2)
relativeCourse = nedCourse - waveCourse
a = tan(relativeCourse)
fFront = Kf/(fCurrent + fWind)
fSide = Ks/(fCurrent + fWind)
c = fFront*waveHeight
d = fSide*waveHeight
b = 0
a0 = (1/(c**2)) + (a**2)/(d**2)
b0 = (-2*x0)/(c**2) + (2*a*b ‘- 2*a*y0)/(d**2)
c0 = (x0)**2/(c**2) + (b**2 - 2*b*y0 + (y0)**2)/(d**2) - 1
solution = quadrticRoots(a0, b0, c0)
v1 = [solution(1),a*solution(1)]
v2 = [solution(2),a*solution(2)]
vCourse = [cos(course),sin(course)]
velocity = max(dot(v1,vCourse ),dot(v2,vCourse ))
```

9.3 Weather

For the simulations three different parameters are of interest. Current, wind and waves, met.thredds.no have many different weather and ocean forecasts. The met.no ROMS NorKyst800m coastal ocean forecasting system gives a 24 to 66 hour forecast on Multiple weather forecasting parameters including wind and current [24]. Forecasts for the current day forecast 66 hours into the future, while, any previous forecasts have a 24 hour horizon. The forecast covers the entire ocean coast of mainland Norway. The forecast has a resolution of 800x800 metre squares for each data point. These forecasts are based on larger global ECEW forecasts generated by European Centre for Medium Range Weather Forecasts[25]. These forecasts have a much lower resolution between around 4000x4000 metre. The Norwegian meteorological institute uses these forecasts as boundary conditions for their own more accurate model ROMS NorKyst forecasts. A similar approach is done for the wave foreacsts which also have a 800x800 metre resolution[26]. The forecasts are however divided into different areas of Norway, namely Finnmark, NordNorge, MidtNorge, Vestlandet and Skagerak. Both forecasts have over 20 different parameters, but the parameters that will be used for this project is wave height and wave direction, current and wind. Other parameters such as wave frequency is also of interest, but is not included because of uncertainty to effect of frequency. A proposal for inclusion of frequency will be made, but not included in the calculations and final results. To simplify implementation and to be able to create forecasts stretching over multiple days, an algorithm was created that stitched together the different 24 hour forecasts into continuous forecasts over multiple days. All current, wind and wave data is combined in a .mat file, to simplify the movement for forecast data and increase the ease of implementing and using Forecast data both for use in this thesis and future use.

9.4 Binary operations

Binary operations can be described as situation where the input of the operation is either 1/0 or true/false, depending on the context. These operations can be a part of the propulsion system like a constant thrust thruster or a description of the state of a sensor or instrument. Optimisation of binary operations does either need its own binary optimisation methods, or a re-description of the binary states to a continuous description. One can for example describe the interval you want the thruster to be. That way a binary operator can be optimised through a continuous optimisation algorithm. This however is not always possible.

During a mission the AutoNaut is expected to perform many actions that can be described as binary operations. Turning on and off sensors, sending messages and enabling and disabling different modes. For this thesis, the sensor usage will be implemented as a binary operation. The optimisation algorithm will therefore be a continuous-binary hybrid optimisation algorithm, that will be able to optimise both the binary and continuous parameters.

9.5 Areas of interest

When gathering scientific data over a prolonged time it is normally desired to take samples at the same location to be able to observe change in the environment over time. It is therefore expected that the mission for the AutoNaut will have some predetermined positions that the vessel has to enter during the mission. The cost function will therefore implement "Areas of interest" which are predetermined positions where it is desirable for the vessel to go and do a specific action. Areas of interest are defined as an area described either by a map, function or just a circle with a radius and origin at a set position. Within this area the vessel is capable of gaining a cost by doing a predefined action. If we set the cost of sampling in a correct area to negative, the vessel will be incentivised to perform an action in the area of interest to minimise the cost function. Another possibility is to implement a hard constraint to the path to force it to a desired point, but in situations where the point is infeasible, the optimisation will not be feasible. Instead, Soft constraints opens for the vessel to skip mission goals if the gain from sampling at a point does not outweigh the cost of reaching it. It can therefore autonomously skip points if it is deemed non optimal.

For this master, the optimisation of sensor usage will be implemented and showcased as a part of the cost function. To simplify the optimisation, it is expected that a sampling will take shorter time than the duration of a time step. Therefore there will only be needed to take one sample inside every area of interest to get the score associated with the point. A more complex description could be implemented if needed. The implemented cost function will describe points of interest as follows:

- Map out N point of interest
- Describe a cost/score to the points and a radius around the points
- Initialise different paths between all points to seed the optimisation algorithm
- Introduce cost to using sensors
- Add constraint to the amount of times a sampling can be done in an area of interest.

9.5.1 Communication

Communication with the AutoNaut normally goes via VHF, 4G cellular or Iridium satellite communication. All these different communication methods have a set of predefined messages that is normally sent at fixed intervals. It is however not always necessary to send redundant messages. When sending message over Iridium you pay for each 50 bytes sent, thus it is sometimes of interest to reserve long messages to when it is necessary. For communication such as cellular, the vessel will need to be within the coverage of the base stations to send a message. By letting the coverage area be the area of interest and the message be the action, a binary operation can be used to describe the message sending in the cost function. The vessel will therefore be incentivised to go to an area with cellular service to send a message, instead of sending it over iridium. Some data also looses value over time. This should incentiveise the cost function to send the data before the monetary value is too low. This is however not prioritised in this thesis, due to uncertainty in practical implementation, and is therefore not implemented.

9.6 Final Cost Function

For the Masters project the final variables that will be implemented will be the running costs of running the vessel and the sensor data. For the moment these are the variables that mostly affects the mission planning. An additional danger cost could also be easily implemented. However, there is no description of danger of loss of vessel yet. For hard constraints it is mostly about constraining the vessel from crashing into land or going into restricted areas. Implementing hard constraints to the parameter space is complex to implement, thus the constraints will be implemented as an infinite cost if a constraint is broken. This also allows for a lot of different nonlinear constraints to be tested out without any need for changing the cost function. The weather forecasts for ocean data has NaN values at areas that cover land. The algorithm will therefore notice if it has crashed into land by checking the forecast data. The 800x800m resolution forecast should give a descent description of land, but the final implementation into a vessel should introduce a refined/conservative map to assure that the vessel will not crash into land. A suggestion for such a map would be to dilate the areas of the weather map with NaN values to assure that the vessel stays away from the shore line. Another potential solution is to use the cross section of land at a certain ocean depth, and that way limiting how shallow waters the vessel is allowed to traverse.

Algorithm 4 Cost Function

```
time = 0
cost = 0
for i = 1:listLength do
    lat = latitudeList(i)
    long = longitudeList(i)
    nextLat = latitudeList(i+1)
    nextLong = longitudeList(i+1)
    current = getCurrent(Latitude, Longitude)
    wind = getWind(Latitude, Longitude)
    waveHeight = getWaveHeight(Latitude, Longitude)
    waveDirection = getWaveDirection(Latitude, Longitude)
    distance = greatCircle(lat,long,nextLat,nextLong)
    speed = vesselVelocity(current, wind, waveHeight, waveDi course)
    time = time + distance/speed
    course = course(lat,long,nextLat,nextLong)
    cost = cost + missionGoals(sensorList(i))
    cost = cost + hasCrashed(lat, long)
end for
cost = cost + time*timeCost
return cost
```

9.7 Particle Swarm Optimisation

To implement both binary and continuous parameter optimisation, a hybrid binary continuous particle swarm algorithm is implemented that is capable of optimising both input types. The implementation is made by having a continuous particle velocity be calculated on the continuous paramours and using a binary velocity calculation be used on the binary part of the parameter input. This is a solution simple to implement and effective in use. It has also been used in multiple articles to find solutions in similar hybrid binary-continuous cost functions, especially withing antenna and phased array problems. [27, 28].As previously described, the input parameters will be both way-points in latitude and longitude, and sensor state for each way-point. The algorithm was implemented as follows:

Algorithm 5 Hybrid Particle Swarm Optimisation

```
n = numInputs
m = numParticles
t = maxIterations
$x_{cont} \in R^n X R^m$
$x_{bool} \in R^n X R^m$
$p_{best,p}, p \in R^m$
$g_{best} \in R^1$
for $k = 1 : t$ do
  for $j = 1 : m$ do
    for $i = 1 : n$ do
      $v_{cont,i,j} = \omega v_{cont,i,j} + \alpha(p_{best} - x_{cont,i,j}) + \beta(g_{best} - x_{cont,i,j})$
      $x_{cont,i,j} = x_{cont,i,j} + v_{cont,i,j}$
      $v_{bool,i,j} = \omega v_{bool,i,j} + \alpha(p_{best} - x_{bool,i,j}) + \beta(g_{best} - x_{bool,i,j})$
      $x_{bool,i,j} = x_{bool,i,j} + v_{bool,i,j}$
    end for
  end for
  for $j = 1 : m$ do
    if $costFunction(x_{cont}[:, j], x_{bool}[:, j]) < p_{best,j}$ then
      $p_{best,j} = costFunction(x_{cont}[:, j], x_{bool}[:, j])$
      $x_{pBestCont,j} = x_{cont}[:, j]$
      $x_{pBestBool,j} = x_{bool}[:, j]$
    end if
  end for
  if $min(p_{best}) < g_{best}$ then
    $i = indexMin(p_{best})$
    $g_{best} = p_{best}[:, i]$
    $x_{gBestCont} = x_{pBestCont}[:, i]$
    $x_{gBestBool} = x_{pBestBool}[:, i]$
  end if
end for
return $g_{best}, x_{gBestCont}, x_{gBestBool}$
```

Part IV
Analytical Tests of Optimisation Algorithm

10 Analytical Tests - Particle Swarm Optimisation

10.1 Time Invariant Current

To test and verify models and optimisation algorithms, analytical tests and simulations are used to verify that the algorithms performs as expected. Both the parameter estimation method and the optimisation algorithms were tested in different cases to verify results and compare the results to similar tests done in previous articles. To test out the algorithms in an analytical scenario the weather data gathered from met.thredds.no [26, 24] was replaced with simulated weather data modified to simulate certain scenarios. To test out the optimisation algorithm, different environments was simulated to challenge the optimisation algorithm.

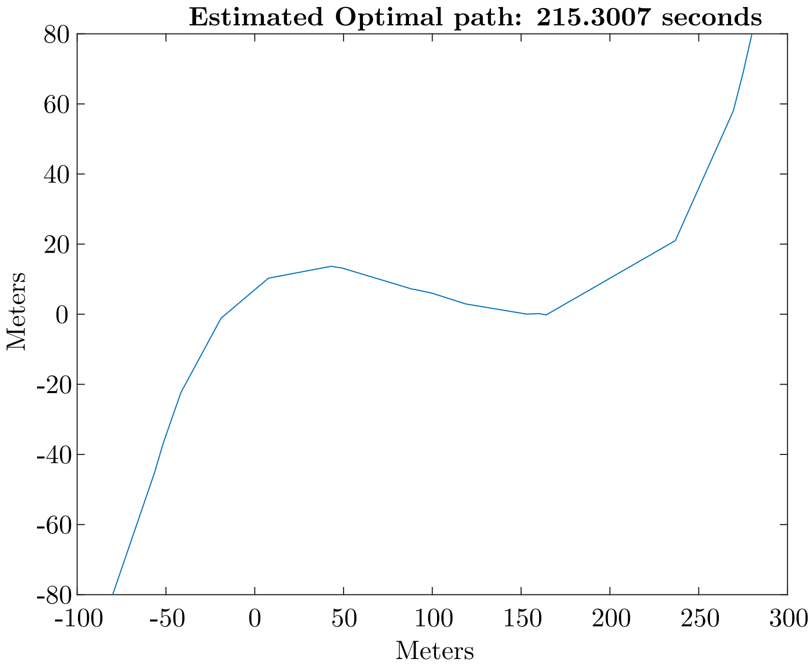


Figure 11: Figure showing how the time optimal path adjusts the path for the current

Kularatne 2016 et al [8]used a repeating vortex current to show optimality of the algorithm. To compare the results comparing an A* algorithm to to the Particle swarm, the first optimisation algorithm will use the same current. The current can be described by a vector field $V_c in R^2$ used is an analytical model

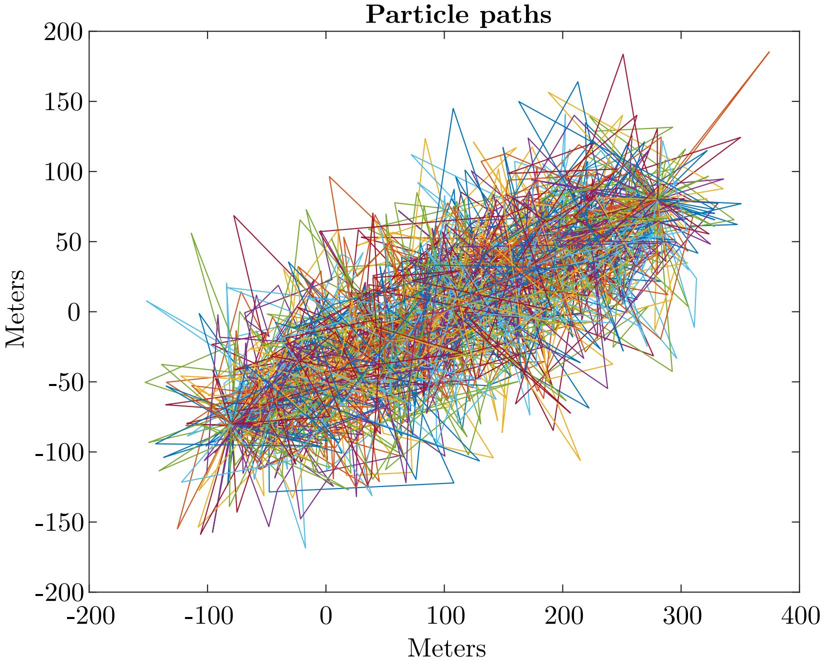


Figure 12: Figure showing how the initial particle paths. Each colour represents one particle. The end result is largely determined by the initial particle states

Parameter	F_f	F_s	D_c	D_w
Value	200	200	100	0

Table 1: Table showing parameters used for velocity function for time invariant analytical current

Parameter	Particles	Iterations	ω	a_1	a_2	a_3	Particle Distribution
Value	100	200	0.2	0.1	0.7	1	40

Table 2: Table showing PSO parameters for time invariant analytical current

with a set of spiral currents. This creates a challenging path where the straight forward path is rarely expected to be the optimal. It is still intuitive to verify visually that the optimisation algorithms takes into account current and wind. To be able to compare to earlier results, the simulations will recreate previous results to verify that the new method can perform as expected and give comparable results to earlier tests, while also implementing new functionality to improve on previous results. In Kularatnes tests the paths were made to optimise paths for small radio controlled vessels only 10 centimetres long. Therefore the scale in the tests was only 3x3 metres. To get a more realistic setting the scale of the test was increased to around 400x300 metres, which is expected to be the upper bound of planning resolution. However, it should be noted that since vessel acceleration and yaw dynamics are neglected, the optimisation result of the scenario will be geometrically similar, thus independent of scale.

To get a repeating vortex pattern vector field current can be described as

$$V_c = \begin{bmatrix} \sin(y) \\ -\sin(x) \end{bmatrix} \tag{60}$$

The vector field V_c has diagonal lines along x plus minus y, which is less intuitive. Therefore we rotate the vector field 45 degrees clockwise, in the current setup. Results should still be comparable to testes in [8] due to acceleration not being a factor in the optimisation algorithm, to increase the similarity between the article and these results. Forces from wind were set to 0 and $F_s = F_f = 200, /D_c = 100$. That made the maximum velocity of the vessel relative to the water is 2 m/s. The force generated by the waves also represent a circle similar to the article. This makes the velocity estimation similar to the article, except for not having adjustable actuator force.

10.2 Results - Time Invariant Current

In the results it can be observed how the optimisation algorithm manages to find paths similar to the results in [8]. The path curves along the currents to take advantage of the increased speed. The results show around a 22 percent reduction in time from 280 seconds to 216 seconds. In Kularatnes tests the vessels had a much lower propulsion velocity, as they were made to model small radio controlled vessels.

You can see from the particle swarm costs in figure 28 that all the particles collectively reduced their costs during the optimisation iterations. You can also observe that the costs for the particles seemed to hold a lower bound in the variance between the particle costs. This mostly comes from the added random velocity added to the particle velocity for each iteration. Using random guesses in discrete path planning often creates jagged non optimal paths, this holds for both Particle swarms and other similar algorithms such as Raplidly-exploring Random Tree algorithms (RRT)[29]. To increase the optimality of RRT algorithms a version of the RRT algorithm called RRT* is used to smooth out and optimise the feasible path. This is done by trying random "shortcuts" in the paths, and saving the new paths that increases the optimality of the path. The same logic can be implemented for PSOs to also reduce the jaggedness of the results.

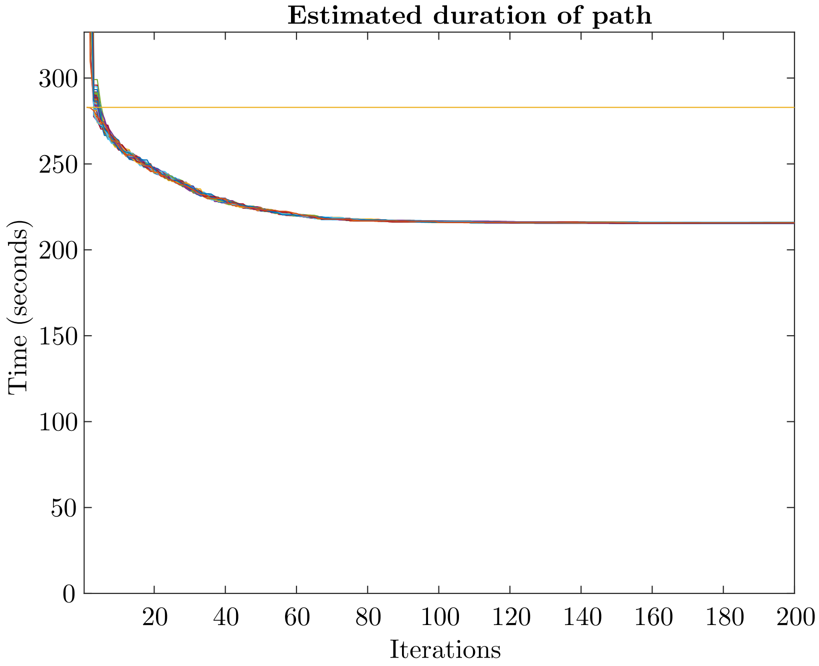
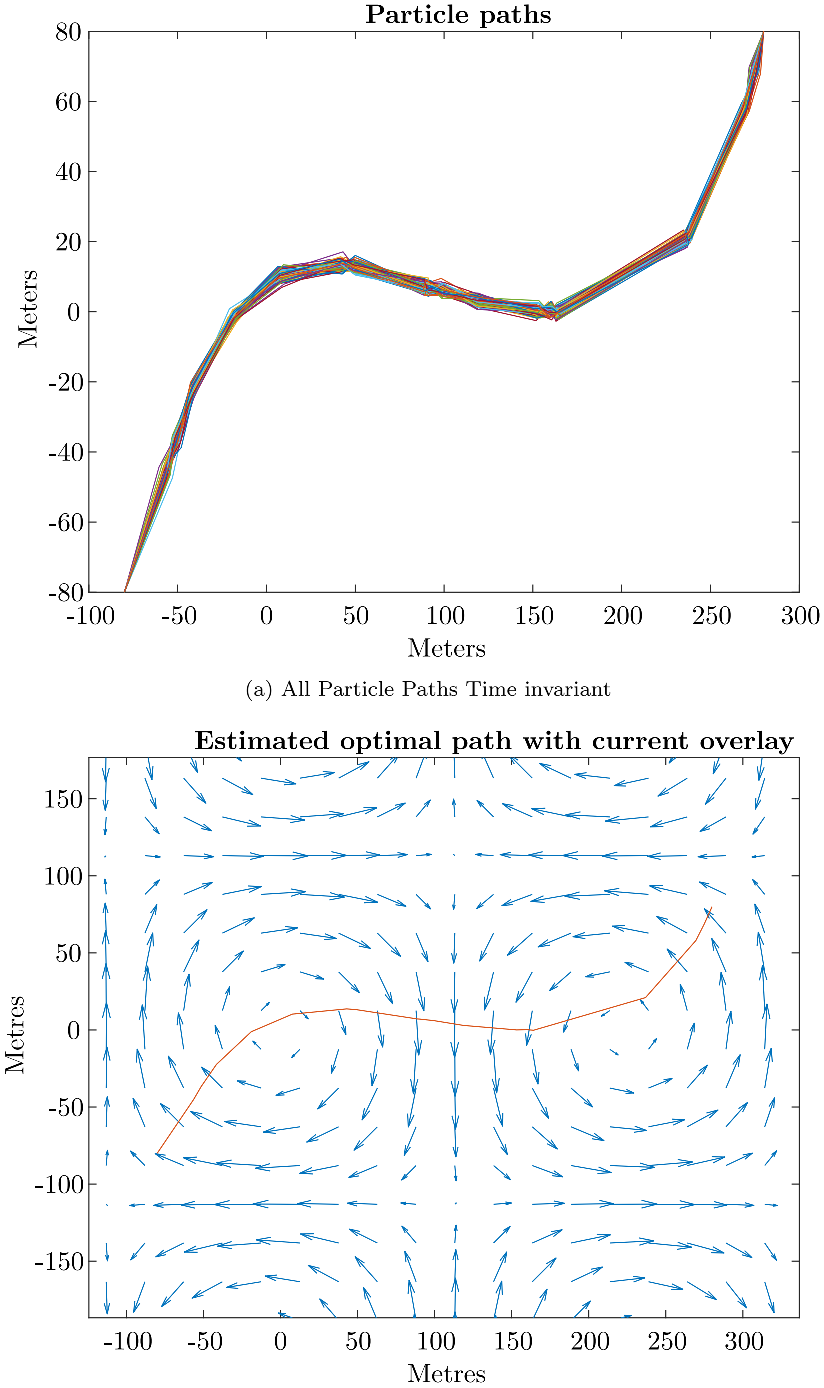


Figure 13: Figure showing evolution of particle costs for time invariant current. Yellow line represents the straight path solution



(a) All Particle Paths Time invariant

(b) Optimal Path with Current Overlay

Figure 14: Three sub-floats.

10.3 Analytical Test - Time Varying

To test out how the algorithm handles difficult scenarios, the tested analytical mission was expanded to include time invariant currents. Under normal conditions the weather is expected to be continuous both in time and space. Many path planners, like time varying A* algorithms [9] and potential field algorithms [12] use the assumption of continuous velocity fields for current to improve the results. However, even if discontinuities are rare, they are not impossible. The discrete nature of weather forecasts can introduce discontinuities into the data used for mission planning. Discontinuity in the weather data will create problems for many algorithms, as an algorithm will not be able to estimate the optimal solution given only local knowledge. A particle swarm however weighs optimality of a particle given the entire mission cost. This should allow a particle swarm to more easily find an primal solution under conditions with discontinuities. Discontinuities are however expected to increase the necessary particle density, as discontinuous environments have a larger tendency to ”hide” potentially beneficial spots.

To test the optimisation algorithm the current will be described by a time varying discontinuous current. The current vector field is made where y ¡ 0 has opposite current of y¡0 where both currents are time varying. To set up a fitting mission, the vessel will have to travel along y=0 line which splits the current vector field. The current will be set to 0.8m/s maximum. This will make the straight forward path inefficient. Ideally the optimisation algorithm should constantly switch bellow and above the Y=0 to take advantage of the current when the current flows towards the goal. The wave is set to 1 metre, the wind drag set to 0 and $F_s = F_f = 200, /D_c = 100$. The distance to be travelled is around 400 meters, which would mean 200 seconds if there is no current. Let x(0) = (0,0) and the end point be x(T) = (400,0). The time varying current field is set to:

$$V_c = \begin{bmatrix} 0.8\sin(0.02\pi t) \\ 0 \end{bmatrix}, y > 0 V_c \qquad = \begin{bmatrix} -0.8\sin(0.02\pi t) \\ 0 \end{bmatrix}, \leq 0 \tag{61}$$

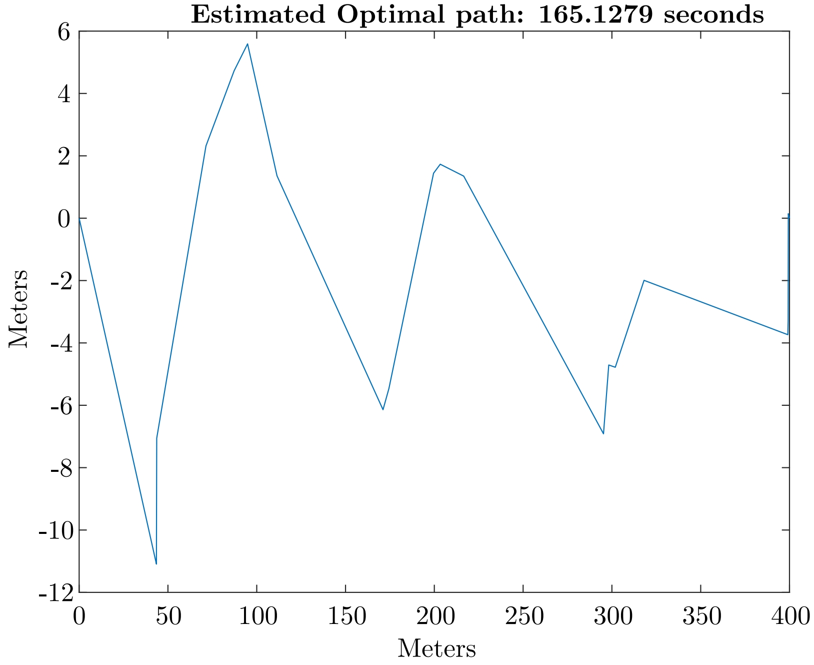
Parameter	F_f	F_s	D_c	D_w
Value	200	200	100	0

Table 3: Table showing parameters used for velocity function for time variant analytical current

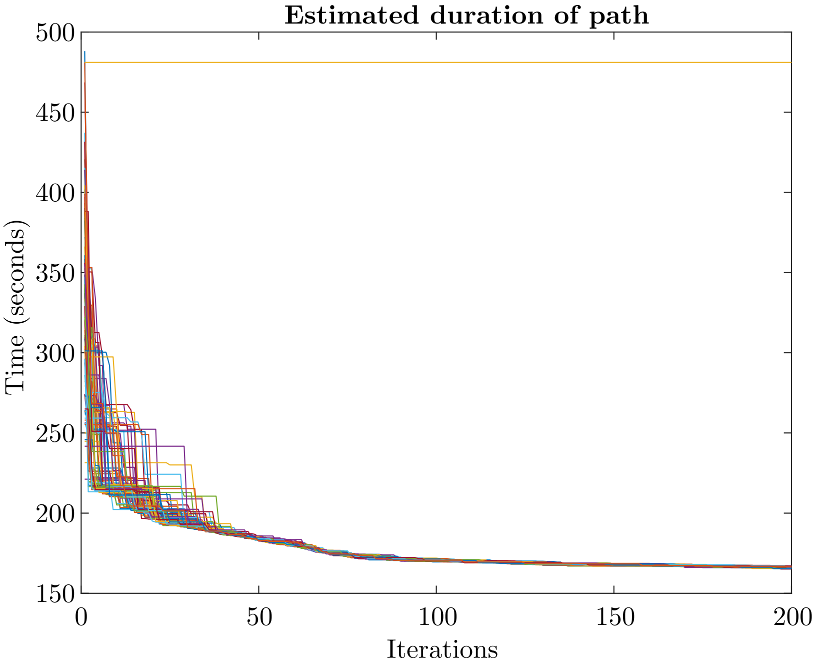
Parameter	Particles	Iterations	ω	a_1	a_2	a_3	Particle Distribution
Value	100	200	0.2	0.1	0.7	1	5

Table 4: Table showing PSO parameters for time variant analytical current

10.4 Results - Time Varying Discontinious Current



(a) Optimal path for Time Variant Current



(b)

Figure 15: Time variant results

In figure 15 you can observe that the path switches below and above y = 0 as the path progresses towards the end position. One of the defects of discretisation can also be observed in the path plots. This most likely comes from a defect of discretising the problem. When the vessel velocity between way-points are calculated, the current is estimated to stay constant between the way-points. Therefore, by matching the way-points with the times when current is at its maximum velocity, the resulting vessel velocity is maximised. This is not representative of the actual system, and are therefore unwanted defects from the discretisation process. You can also see that the next to last way-point is not evenly spaced between the last and the second to last way-point. This is expected to be a similar effect where the optimisation tries to skip the non optimal current flows. These defects do not seem to create unreasonable results, and is considered to be within acceptable error margins. A possible solution to fix the problem would be to add intermediate steps between the adjustable way-points to estimated vessel velocity. It is worth noting the axes dimentions in figure 15 as the x axis is 400 metres long and y axis only 18 metres. The path therefore looks much more jagged than the real result. In the Estimated duration you can see how the algorithm managed to reduce the path duratin from 489 seconds to around 166 seconds. Reducing the path duration to less than half

It is worth noting that the particle swarm optimisation algorithm does not need any changes to the optimisation algorithm when going from a time invariant to time variant system. Only the cost function describing the system needed to change some variables from time invariant to time variant. This is a large pro when comparing the optimisation algorithm to an A* algorithm where the neighbours of the vertexes have to be calculated for each timestep. parameters[9] which needs extra complexity to be able to solve.

Parameter	F_f	F_s	D_c	D_w
Value	200	200	100	0

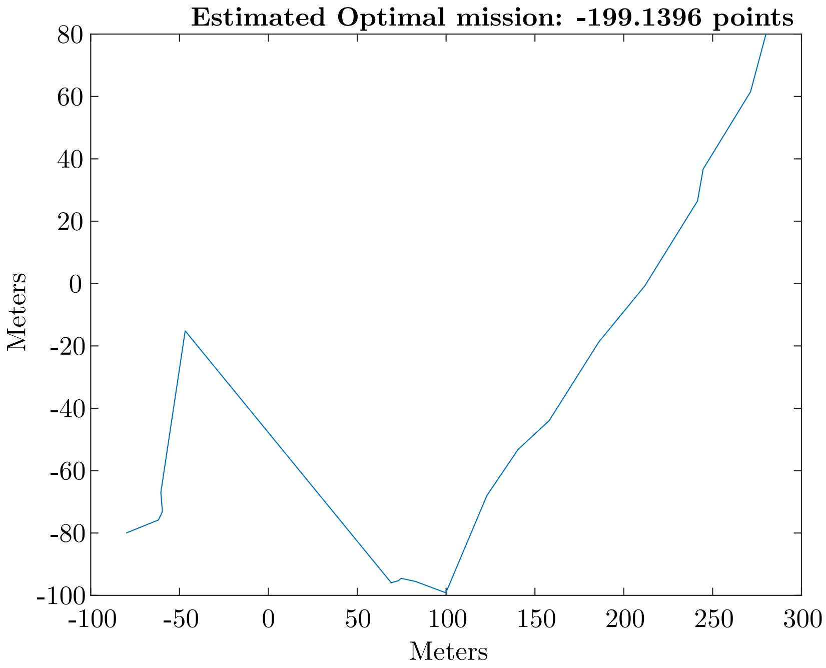
Table 5: Table showing parameters used for velocity function for analytical mission with area of interest

Parameter	Particles	Iterations	ω	a_1	a_2	a_3	Particle Distribution
Value	100	400	0.2	0.1	0.7	1	40

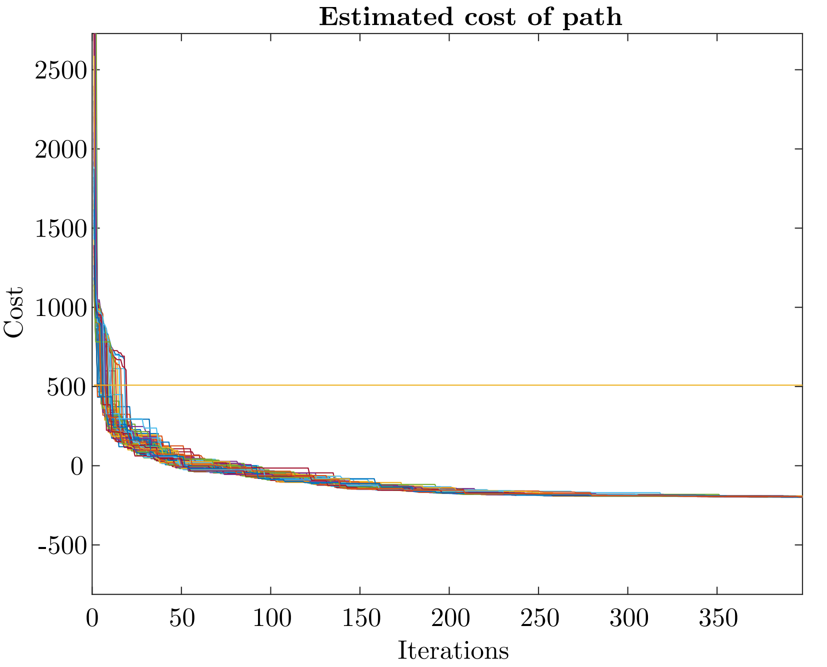
Table 6: Table showing PSO parameters for analytical mission with area of interest

10.5 Binary Decision Optimisation

The unique part of the optimisation algorithm that separates it from previous algorithms, is the addition of areas of interest. In these areas the vessel has an added functionality to set a parameter from 0 to 1, representing a sensor being turned on. This allows the optimisation algorithm to optimise both the path and the sensor usage. The need for being conservative with sampling might not be intuitive. However in situations where all samples are to for example be sent over satellite communication, the cost of transferring sample data might be a large part of the mission budget. In this simulation the cost of using the sensor is set to 20 for each way-point where the sensor is on. If the vessel manages to turn the sensor on withing the area of interest situated at (100, -100), the mission cost will be reduced by 500. The cost for turning the sensor on was set to 20. There is also 1 cost added for every second of mission duration. All other parameters were set similar to the analytical time invariant current test, except for iterations, which were set to 400 to allow for the particle swarm to set all sensor states.



(a) Optimal path for mission w/area of interest



(b) Particle mission costs w/area of interest

Figure 16

10.6 Results - Binary Decision Optimisation

When looking at the results in figure 16, One can observe that The algorithm is capable of finding a path that also includes the area of interest at (100,-100). It can however be seen that the optimal path has large jumps and spikes similar to results with the discontinuous time variant current. The algorithm managed a reduction of mission cost from 500 to -199 16. Since the cost does not represent time anymore, the optimal cost can be negative.

10.7 Analytical PSO Tests -Conclusion

For both Time invariant currents , time variant discontinious currents and missions with areas of interest. The algorithm was able to find solutions to the problem, more optimal than the naive straight line paths. This shows how the PSO is capable of handling situations pushing boundaries. However, it can also be seen that the algorithm shows undesired behaviours. As a proof of concept however, the results are deemed sufficient.

Part V

Parameter Estimation of Real Mission Data

11 Real life Parameter Estimation

11.1 Parameter Estimation - Trondheim Fjord

The main base of operations for the AutoNaut is Trondheim Biological Station (BTS). The station is a part of NTNU, having their main focus on marine research. From this station the AutoNaut has been launched, an surveyed from while doing tests in the Trondheim fjord

The Trondheims fjord is Norways third longest fjord stretching around 126 kilometres. The opening of the fjord is around 4 kilometres wide. BTS/Trondheim is situated around 50 kilometres inland of the "fjord opening", thus the area of operations is mostly sheltered from bad weather. There is around a 10x10 km open area straight outside Trondheim city which is where much of the testing has been carried out. The large open space and currents in the fjord allows for large enough waves for the AutoNaut to propel itself. The Trondheim fjord makes great conditions for testing the AutoNaut in onshore conditions where there is traffic from both heavy supply ships and leisure craft. In previous experiments, avoidance algorithms for the AutoNaut have been tested here.

11.1.1 Conditions

The date of testing was the 20. February 2020. Temperature were around 0 degrees. Waves were between 0 to 1 metre high. There were little to no current between 0.3 -0 m/s . These waves were sufficient to propel the vessel at an average of 0.43 m/s The AutoNaut Started in the Trondheim harbour going straight north before following a rectangle path to get wind current and waves from all angles. The vessel drove in a square in the Trondheim's fjord before finishing the trip back.

11.1.2 Estimated Parameters

When estimating parameters there are a lot of different decisions that are capable of affecting the results. Measurement sensors, sampling rate, measurement noise and vehicle path will all have some effect on the parameter estimation. The AutoNaut is a continuously developing vessel and the sensors aboard are continuously changing. The sensors on the AutoNaut have therefore undergone changes between the tests in Trondheim and Mausund. The amount of sensors used has therefore been reduced down to vessel IMU and GPS. Via vessel GPS the positioning of the vessel has been sampled at a 2 Hz frequency. The vessel IMU has been sampled at a $\frac{2}{3}$ Hz frequency. The sampling rate is not synconised either. The heading data was therefore linearly interpolated to get a matching heading data for each GPS position measurement. The momentum of the vessel and friction from the ocean means that the linear interpolation likely were a sufficiently adequate estimate for vessel heading. Any invalid data in the data sets were removed. Vessel velocity was estimated by dividing the Great circle distance between two way-points measured by the GPS. The acceleration was similarly estimated by dividing the change in velocity by the time between each velocity estimate. One issue that stands out is that for high sample rates higher than the wave frequency, the sampling will measure acceleration caused by the waves rolling and pushing the vessel. These movements are not implemented into the current model. These forces will therefore act on the parameter estimation as noise. The estimated results should therefore be considered as a proof of concept rather than an accurate method of parameter estimation.

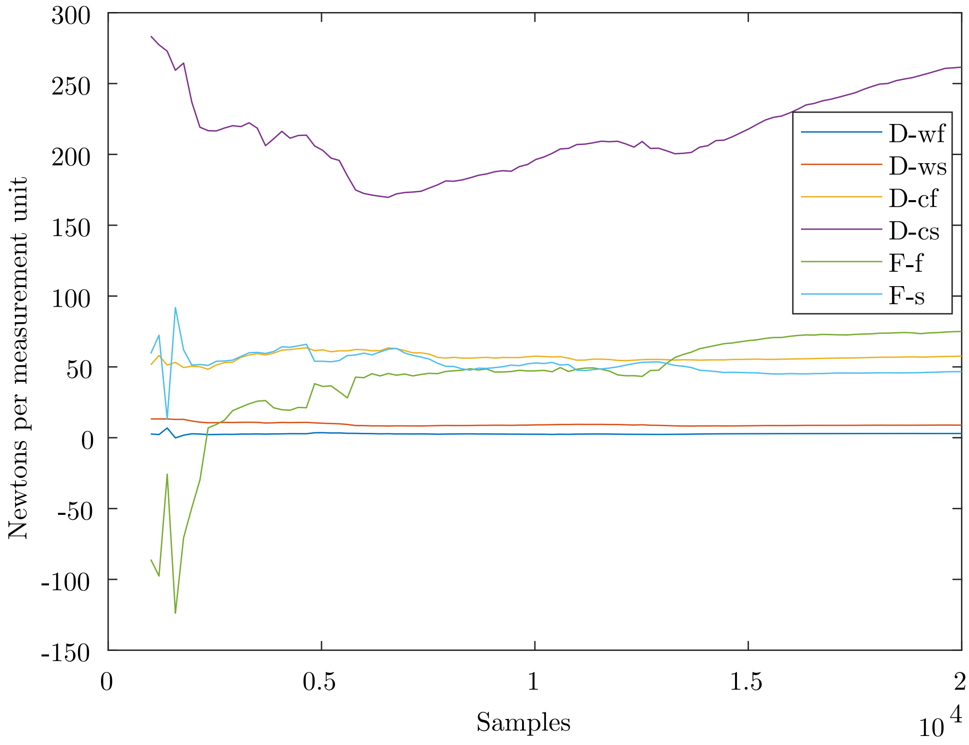


Figure 17: Figure showing estimated vessel parameters for half second sample rate, estimated in a test in the Trondheim fjord 20/02/2020

11.2 Parameter Estimation Result

Three different graphs were included showing three different parameter estimates. For the three different sampling rates it can be observed that the parameter estimations is similar for most parameters at 2 Hz and 1 Hz. At 0.5 Hz however it can be observed that all the estimated parameters drop to a third of the 1 Hz estimates. Decreasing the sampling rate even further drops the estimated parameter values even further. The reason for the reduced vessel

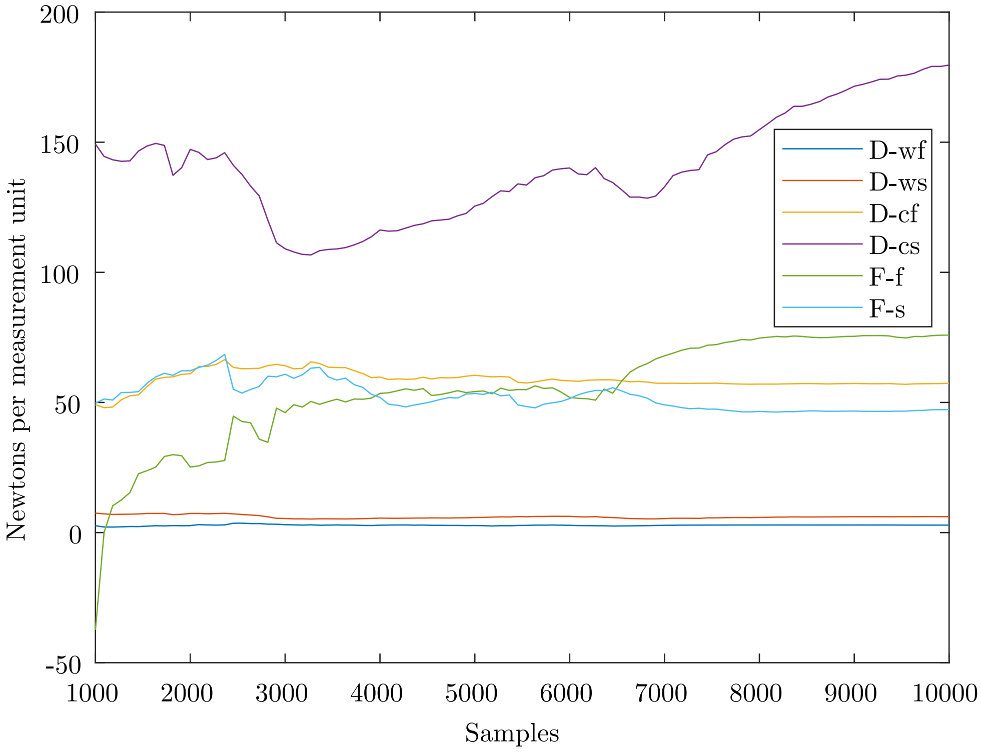


Figure 18: Figure showing estimated vessel parameters for one second sample rate, estimated in a test in the Trondheim fjord 20/02/2020

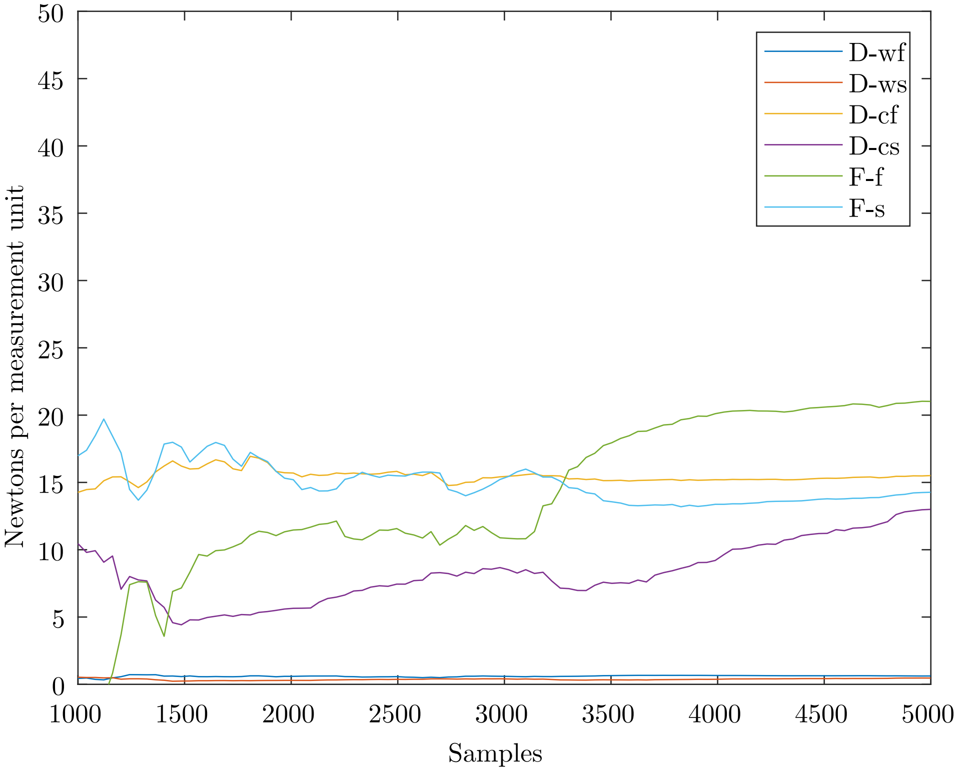


Figure 19: Figure showing estimated vessel parameters for two second sample rate, estimated in a test in the Trondheim fjord 20/02/2020

parameters is most likely caused by that the sampling rate is too slow to accurately describe vessel dynamics given our model. Another way of interpreting the results are that the estimated acceleration is largely affected by roll and pitch of the vessel. As the sampling rate went down, the contribution of acceleration from the waves were reduced and the resulting values were generated from actual acceleration. This gives interesting data regarding the frequency response of the vessel, but more accurate models are needed to give a more refined image of the vessels dynamics. Another interesting observation is that the estimated sideways drag is largely dependent on sampling frequency. As the vessel is created to move efficiently forward, the expected forward force is expected to be drastically less compared to the sideways drag. It should also be noted that since the Hydrofoils are only creating a forward force, you will only get sideways body acceleration from change in wind or current relative to the vessel body. The hourly resolution of the weather forecasts resolution the also makes it hard to trust, when the sampling rate have over a 1000 fold higher time resolution than the forecasts. A more accurate estimate could be made with on board wind and current sensors.

As the graphs shows there are clear differences between the forces generated from wind and current. On average the forces created from waves are shown to be around a tenfold less. Previous algorithms normally disregard wind when estimating the weather effect on vessel dynamics. This assumption does "probably" good in good weather conditions, but the estimated drag coefficients show that the forces from wind should not be disregarded for planners wanting to plan for offshore or long duration planning.

When looking at the drag forces It does however seem constant for all angles. Implying that the estimated current forces are a good representation, at least for the conditions during the test. The estimated wave propulsion force does however seem to differ during the test. Comparing the wave propulsion forces with vessel heading implies that the shape of the oval differs from the estimated pseudo oval. A future iteration should try to make a more correct estimate for this force, given a similar model.

11.3 Tests in Mausund

Mausund is a small island at the edge of the Atlantic ocean. There is little to no shelter form the weather and large cargo and supply ships, as well as fishing vessel daily pass trough the area. The area has "a lot" of bad weather.

Mausund has a biological research base in the area allowing the vessel to be launched an retrieved without having to leave the area. These conditions are similar to the expected offshore conditions the AutoNaut is expected to meet while out at sea. The AutoNaut has been tested at multiple occasions out at Mausund and has been spanning days out at sea without human intervention. This alone has been a large milestone for the AutoNaut project at NTNU.

To estimate the vessel parameters for offshore environment, the parameter estimates was run with similar setup to tests done in Trondheim's fjord. The sampled data were taken from tests a few kilometres offshore of Mausund 07.03.2020 and 17.06.2020. Both tests having around 0.5-1 metre high waves. The current were around 0-2 m/s respectively.

11.4 Mausund Results

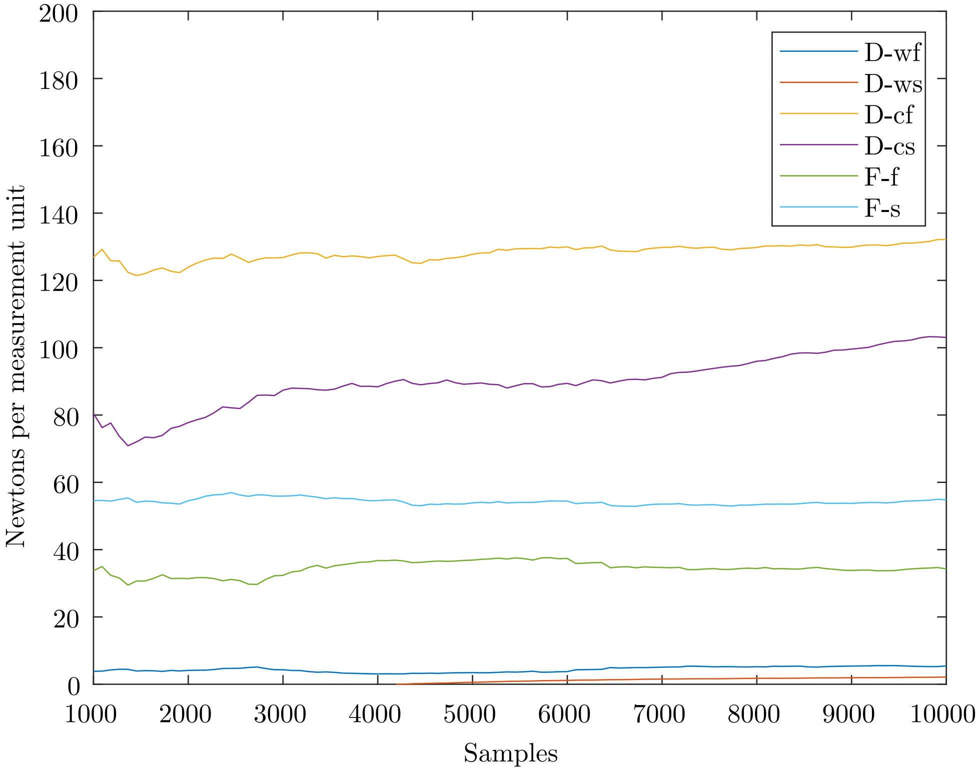


Figure 20: Figure showing estimated vessel parameters for one second sample rate, estimated in a test in Mausund 17/06/2020

In the figures representing the vessel parameters you can spot that the results have definitive differences to the results from the Trondheim's fjord in February 2020. Both results estimate a much higher forward drag, but in the ballpark of similar Hydrofoil forces. It is however interesting that the sideways hydrofoil force was calculated to be stronger than forward hydrofoil force for the Mausund test at 17/06/202020. These tests also had similar sampling frequency depen-

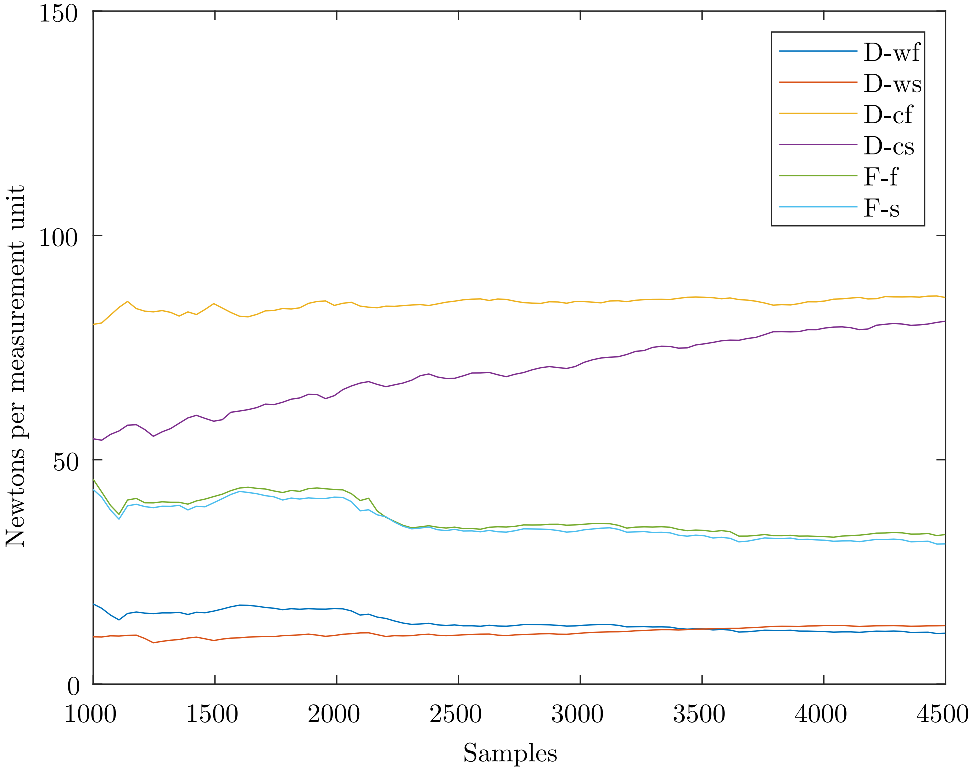


Figure 21: Figure showing estimated vessel parameters for one second sample rate, estimated in a test in Mausund 03/07/2020

dency in estimated parameter forces where the estimated forces reduced with a reduced sampling rate. The average velocity however was about 50 percent faster than tests in the Trondheim Fjord. One might suspect that the increased drag comes from quadratic drag forces. However, due to previous comments about doubts about the method, these results should be used more as a guide for further development, than a benchmark for actual vessel parameters.

Part VI

Test of Mission Planner With Real Forecast Data

12 Optimal Mission Planning using real Weather Forecasts

To test out the vessel planner, a short mission in the Trondheim's fjord is tested out. Vessel will start a few kilometres outside the Trondheim harbour. The vessel will then go to the western part of the fjord where there is a sample point. After reaching the western sample point, the vessel will head east to the Eastern sample point before having to return to the start point. The weather that will be used, is the similar forecast to the forecast used estimating the vessel parameters in the Trondheim's fjord 20/02/2020 18. The mission is similar to short range missions previously performed to test The vessel. Each area of interest gives a 5000 reduction in mission cost, and each use of the sensor gains 100 to the mission cost. Each area of interest can only be sampled one time.

Parameter	F_f	F_s	D_c	D_w
Value	200	200	100	0

Table 7: Table showing parameters used for velocity function for analytical for real life tests in Trondheim

Parameter	Particles	Iterations	ω	a_1	a_2	a_3	Particle Distribution
Value	100	200	0.0004	0.1	0.7	1	0.005

Table 8: Table showing PSO parameters for real life tests in Trondheim

12.1 Optimal Planning Issues

During testing, one main issue was not possible to overcome, using the current particle swarm optimisation methods. When using estimated vessel parameters in particle optimisation, the particle swarm was not able to find any feasible path. The time varying current and wind forecasts seemed to be too complex to be able to find any start path to start the optimisation. To have a closer look at the challenge, a grid test was done to map out the feasibility of moving in different directions. A 30X30 grid between 63.458249 - 63.577489 degrees latitude, 10.292981 - 10.585492 degrees longitude. For each point in the grid the estimated velocity for going North, East, South and west was calculated. These plots together with the current, wind and wave height plots are included in figures 31 to 37. The estimated velocities are made by using weather forecasts at 20/02/2020 at 12 noon. From the different plots it can be observed that all directions have dead spots, or close to dead spots where the estimated velocity is 0 or less. Any non positive or unfeasible area is set to 0 to make the graph easier to understand. These dead spots often cover almost the entire fjord, therefore making it hard for the path planner to find a path to ”penetrate the current belts”. In theory the use of random deviations from the straight line path should be able to find a path given infinite particles and large enough spread. However, due to the limited particles used the algorithm has more limited capabilities.

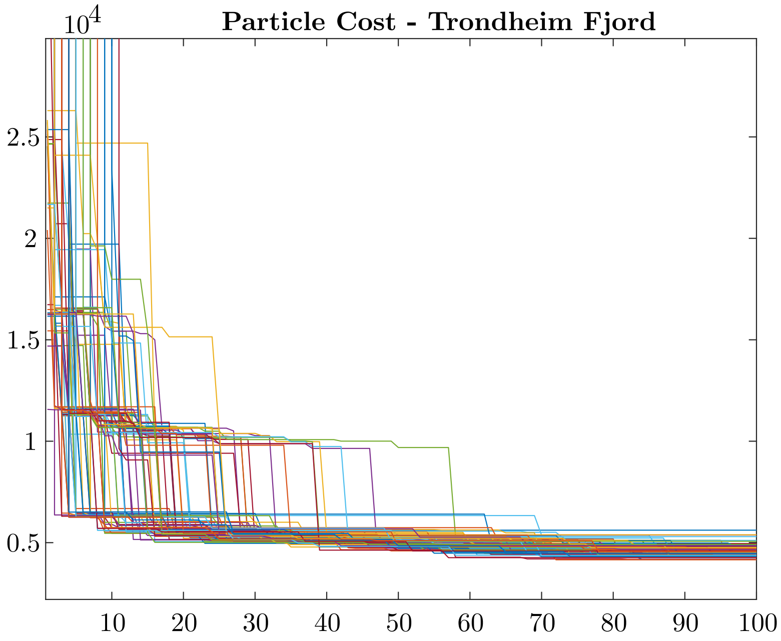


Figure 22: Figure showing the evolution of particle costs for every iteration. When used in real weather situations the results seem to have a much more sporadic behaviour than in the analytical case

12.2 Real Forecasts - Results

For the initial tests with estimated parameters the particle swarm was not able to find a feasible solution. None of the initialised particles were able to find a path that was possible to complete. This issue mostly comes down to the large areas of unfeasible paths, where the vessel can not traverse in the desired direction. During tests in both Mausund and Trondheim fjord the vessel had issues following the preplanned path without using the thruster. It is therefore not unrealistic that the vessel can have problems completing the planned paths. Deciding if the problems stem from unrealistic parameters or realistic dynamics, is hard to decide without testing the path with the vessel. Since the PSO initialises all particles around a straight line path between starting, intermediate and end points, all particles will be initialised in about the same area. If the straight line path crosses a large unfeasible area, then there is a possibility that none of the particles will find a feasible solution. To Simplify the problem, the force parameters for the forward and sideward hydrofoils were increased to 200 forward and 150 sideward respectively. Even if this is not a representation of the vessel it will still give a proof of concept for particle swarm algorithm for

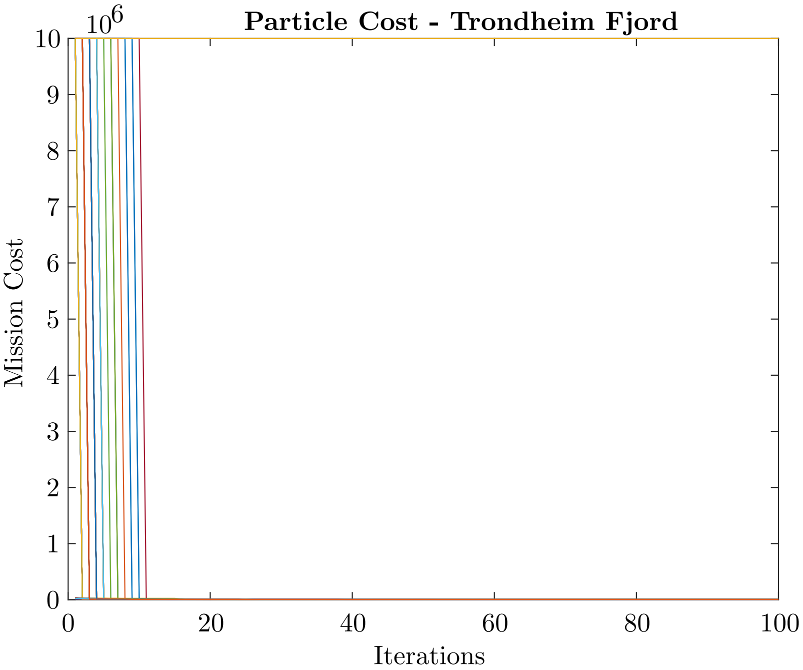


Figure 23: Figure showing evolution of particle costs for every iteration. In this figure, the graph is drastically zoomed out to get a better view of how the particles work. Notice how some particles move from the max value (9999999) down to the other particle costs as the unfeasible particles become feasible.

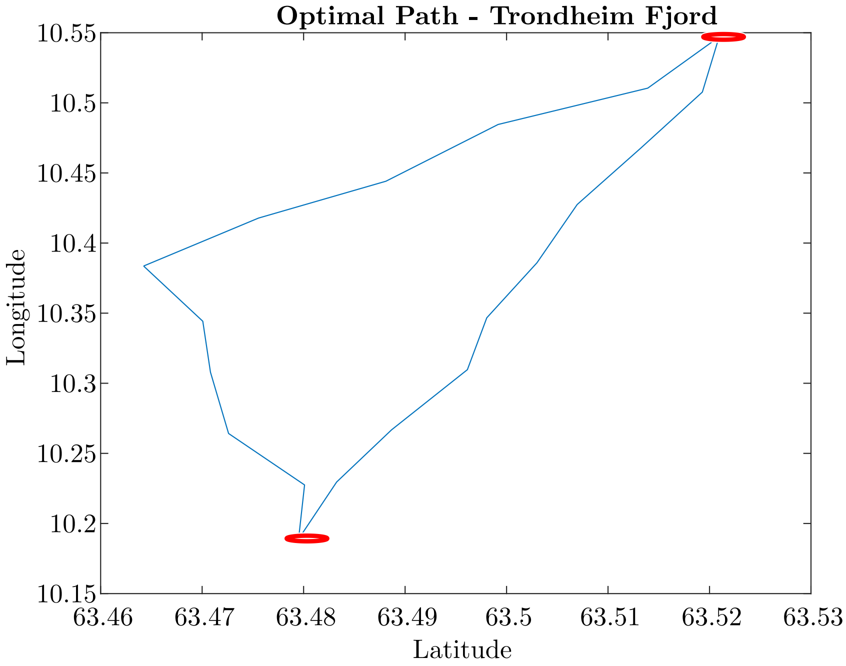


Figure 24: Optimal path for real forecasts. The red circles are areas of interest.

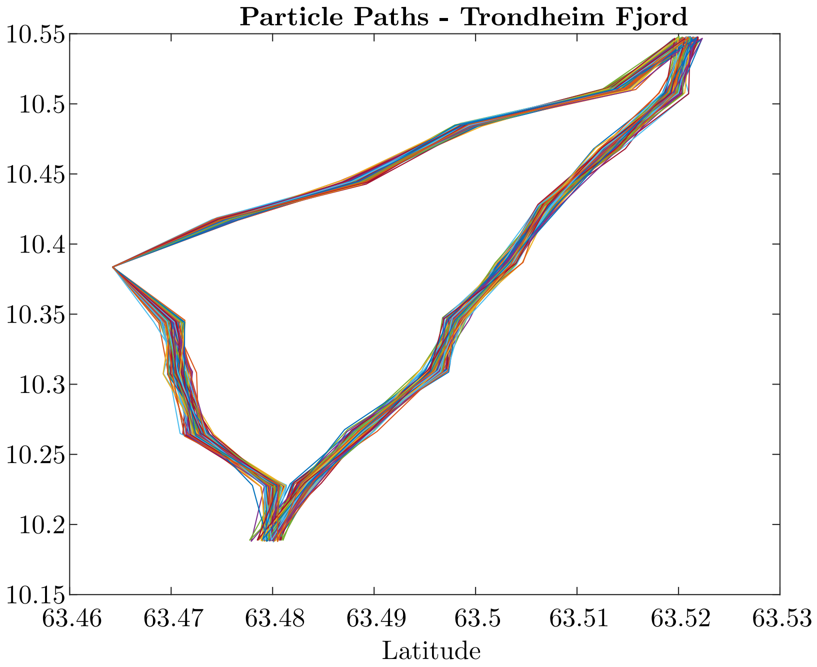


Figure 25: All particle paths

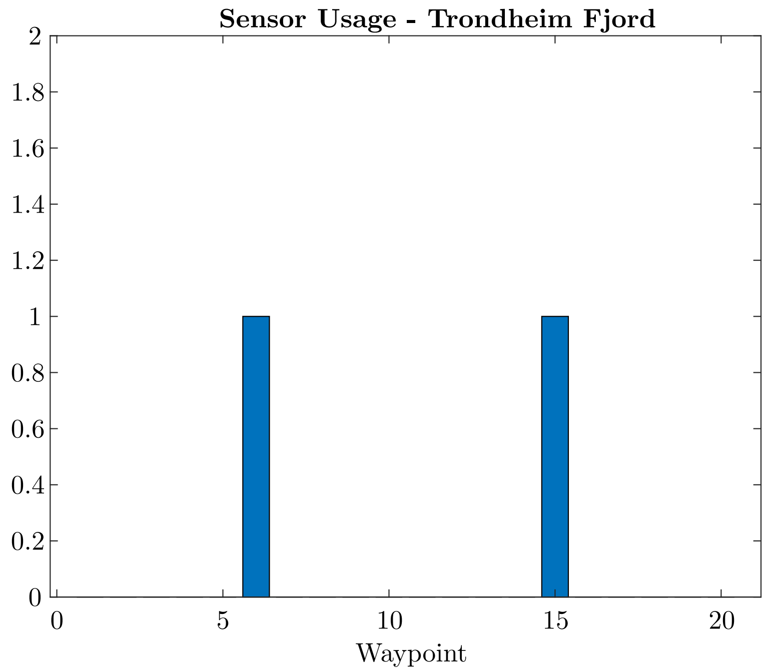


Figure 26: Sensor usage for each way-point.

vessel path optimisation.

When increasing the Hydofoil forces the straight line path was still not able to follow the path, but multiple particles found a feasible solution. When looking at the particle costs over time one can see that the algorithms slowly converged to some possibly optimal solution. When looking at the optimal path it can also be seen that the algorithm did not follow a shortest path, but instead took turns to take advantage of weather effects. The path however is still relatively straight. This is most likely due to the large forces estimated to be generated by the hydrofoils. If the velocity relative to current and wind becomes large, the benefits from following current decreases. The resulting path will therefore be more straight forward than what might be expected. When adding in a time varying current the predictability of an optimal path further decreases. In the zoomed particle cost figure 22 you can see how the particles tend to stop at different levels. These levels are created by the area of interest sensor costs.

When looking at the sensor usage, it can also be seen that the Algorithm only chose to turn on the sensor when in range of the area of interest (red circles). The algorithm was therefore capable of finding a path taking advantage of current and adapting sensor usage to the minimum. As the A* algorithm is generally not capable of handling negative edges, a graph search method would have to use the Bellman Ford algorithm instead, further decreasing the efficiency of graph search. The large negative costs of the areas of interest would also throw off any heuristic guesses, as one would have to include the possibility of drastically reducing the cost by travelling within an area of interest.

It should also be mentioned that the algorithm is capable of starting and ending at the same spot. This is beneficial for situations where the vessel is regularly expected to return to base for retrieving samples or maintenance. A normal shortest path algorithm would normally have to split up the optimisation path to allow this, thus not being able to calculate a globally optimal solution, but only piece wise optimal.

13 Time Invariant Mission Planning

In many of the articles reviewed in the state of the art section, the weather was constant- with respect to time. To be able to create a benchmark to more easily compare the particle swarm optimisation to previous algorithms, a second simulation similar to the time variant test was conducted with constant weather set to the weather from 20-02-2020 at 12 noon. All other parameters were set similar to time varying forecasts.

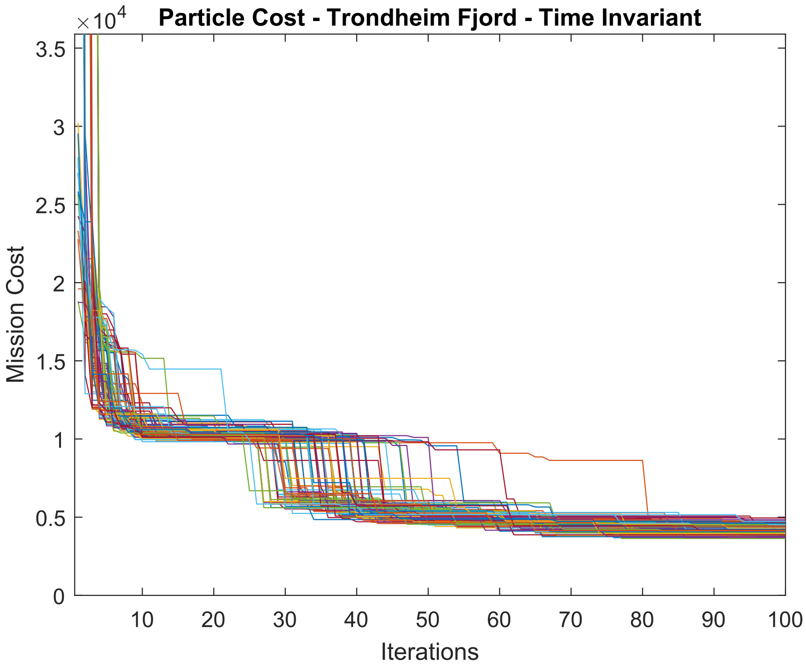


Figure 27: Figure showing the evolution of particle costs for every iteration with time invariant forecast.

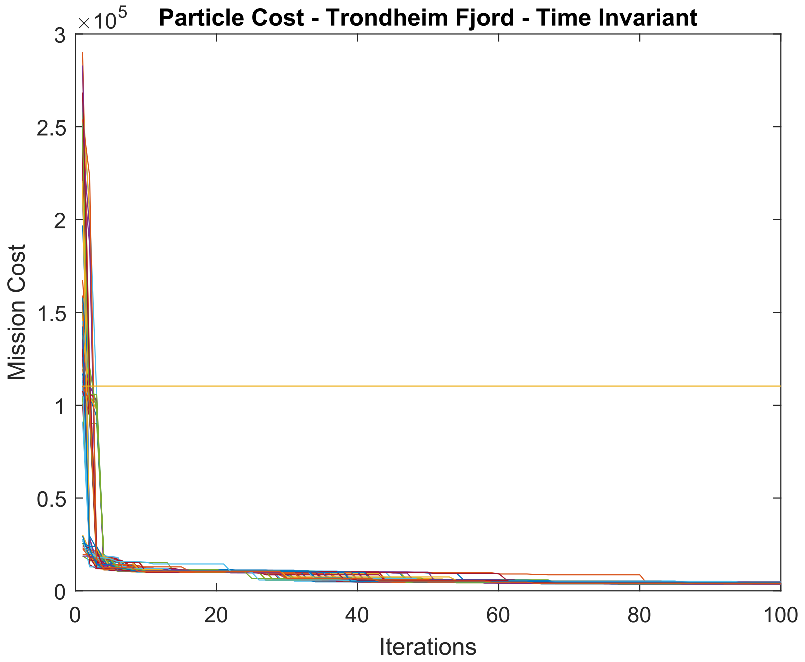


Figure 28: Figure showing evolution of particle costs for every iteration with the time invariant forecast. In this figure, the graph is drastically zoomed out to get a better view of how the particles work. Straght yellow line represents straight path cost

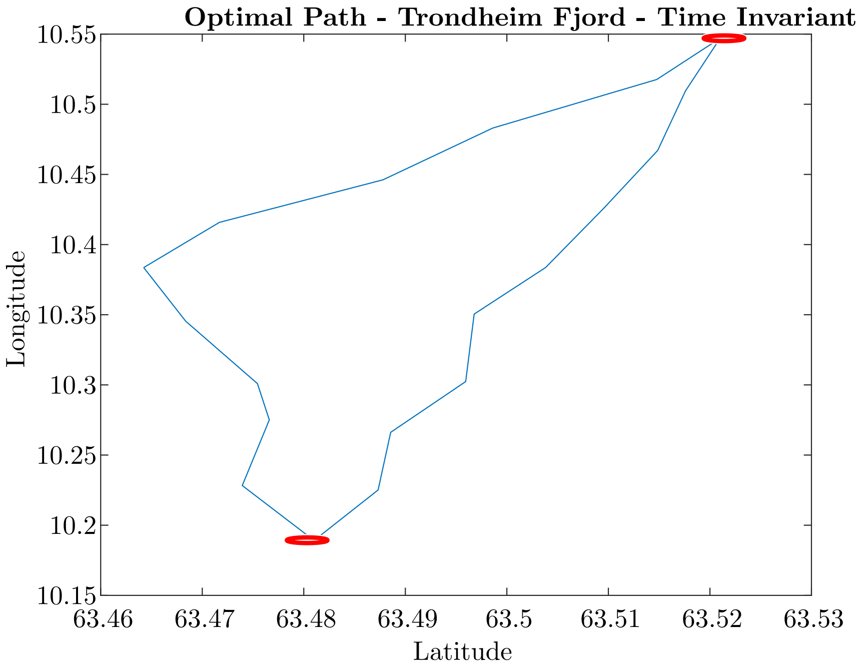


Figure 29: Optimal path with time invariant forecasts. The red circles are areas of interest.

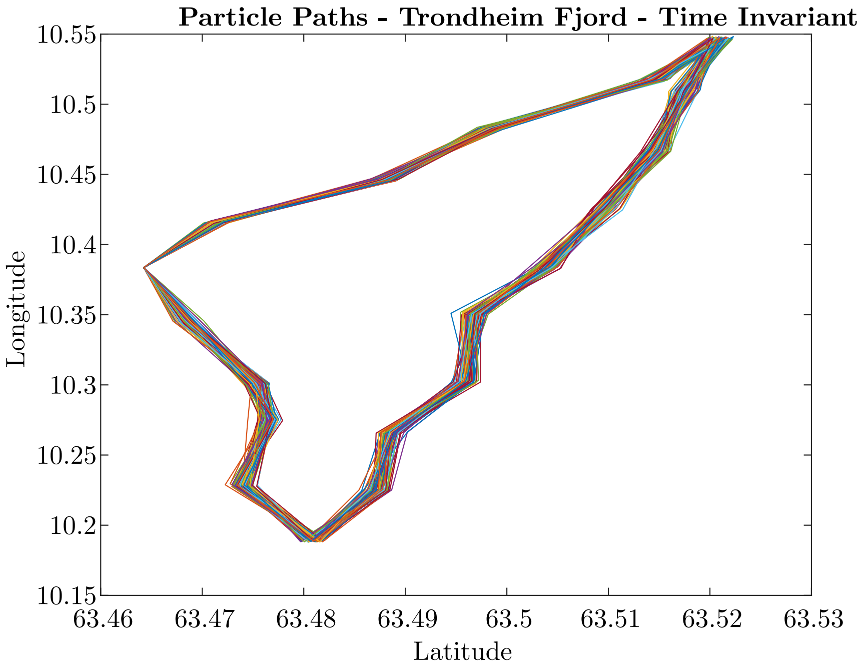


Figure 30: Figure showing the end result for all particles

13.1 Real Time Invariant Forecasts - Results

The optima path found for the Time invariant forecasts displays many of the same properties as the time variant forecasts. It does however seem to take slightly more advantage of currents at some areas. None of the paths however seem to have any drastically different results. The time invariant system were also able to sample within both areas of interest.

Part VII

Discussion and Concluding Remarks

14 Discussion

During this master project it was attempted to create a start to finish mission planner including both optimal paths as well as adaptive sampling. To achieve this, both a mission cost model, vessel model and an optimisation algorithm was created to be able to achieve the end result. The end results shows promise, but the path to a final implementation still needs work. The further work can be divided into three different parts.

For all the analytical results it was proven that the optimisation algorithm is capable of solving the problem the algorithm was designed to tackle. For the mission cost model, there is still not any model for communication, or for risk of vessel loss. Ideally, the vessel should be able to estimate the risk of a mission, to be better able to preserve itself without human intervention. To achieve this, the vessel will also have to prevent the risk of running out of energy. For the vessel to be left without supervision, the vessel will have to assure that it will not run out of energy before heading back home. The cost of communication can also become a large expense if the vessel uses exclusively satellite communication.

The vessel model also have room for improvement. The hydrofoil model was based mostly on observations. A thorough inverstigation into hydrofoil dynamics would greatly benefit the parameter estimation. If the vessel is not expected to face too harsh conditions, a pure current wave model might aslo be a better fit. The parameter estimation could also be further improved by using on board sensors to measure weather instead of using forecasts with one hour resolution.

One path that would especially benefit from further work is the optimisation algorithm. Currently the optimisation algorithm is not good enough at traversing difficult terrain with unknown unfeasible areas. In figures 31 to 34. the feasibility of path directions was mapped out to create a better understanding of the issue preventing the optimisation algorithm from finding a possible path. A grid of points was evenly spread over the fjord outside Trondheim city. Four different graphs were created describing the feasible vessel NED velocity for each point in the grid in the four different cardinal directions, North, East, South and West. The dark blue/purple areas are areas where it is not feasible for the vessel to move in the direction specified in the figure title 31 and onwards. Graphs representing current35, wind 36 and wave height 37 are also included to give a better understanding of how wind, current and waves affects feasibility of paths. Feasibility maps for the increased hydrofoil forces were also included, to show the difference between the estimated parameters and the parameters used in the optimisation algorithm. From figure 38 to figure41 the increased feasibility maps are calculated from the same weather forecast. The feasibility map was created by using the estimated parameters from 20. February 2020. The weather data used were the weather forecasts for 20. February, 2020, 12:00.

From the data in the different feasibility graphs, it can be observed that there can often be large areas where it is not feasible for the vessel to travel in the cardinal direction. The areas feasible are also not convex, creating possible traps for greedy algorithms only focusing on local conditions, for example potential fields. These feasible areas will also be time varying, increasing the difficulty of creating good heuristic estimates for feasible paths. The current method of using the straight line path between way-points has many problems when being faced with unfeasible areas, as the particle swarm only searches for a predefined area around the heuristic guess. In theory you could broaden the particle search, however, particle swarms needs a certain density to prevent the optimisation algorithm from overlooking good solutions. An increased search would therefore increase computation time. A good place to start would fore instance be to implement an RRT algorithm to find a feasible path or multiple feasible paths, which then could be used as a seed for the optimal path planner to find a more optimal path. This would both increase the robustness of finding a feasible path and assure a pseudo optimal path with adaptive sampling.

Currently the algorithm uses the ocean forecast to figure out where land is, as areas of land has NaN values at areas covering land. This does create a correct map, but the way the optimisation algorithm works, it tends to create problems. To find an optimal path, the algorithm wants to "cut corners" to try to find the shortest path possible. This sets the optimal path very close to the shore. This is both a unsafe path for the vessel to try to follow, but also problematic, as the algorithm searches in the area around the most optimal path to try to find other paths. The result is that the algorithm tries many paths that crashes into land giving the particles infinite cost. This is not directly an issue, but each particle that creates an unfeasible path uses computing power without any benefit to the algorithm. Therefore the algorithm becomes inefficient and unsafe close to shore. It should therefore be a more conservative map representing the shore, that prevents the vessel from coming within a predefined distance close to shore. Paths being too close to shore on the conservative shore map but not the weather map, could add a penalty cost instead of an infinite cost. That way less particles becomes useless. It would also allow the algorithm to cross narrow straights that are technically not allowed on a conservative shore map.

When comparing results of the optimisation algorithm to related work, there are situations where the algorithm performs better and situations where the algorithms performs worse. The implementation of sensor usage in the mission cost does allow for a more dynamic sensor sampling. In the current tests, the sensor areas have a binary dynamics. The expansion of a continuous sensor cost model further expands the possibilities of the mission description. When looking at the robustness of the algorithm, previous A* algorithms seem to show a more robust behaviour[9] The A* algorithms have a good capability that given certain constraints the A* algorithm is guaranteed to find a globally optimal solution. Adapting an A* algorithm to work with the models and mission cost adapted for graph search could create a much more robust planner, but at the cost of a larger computational cost, as well as new issues that had to be solved.

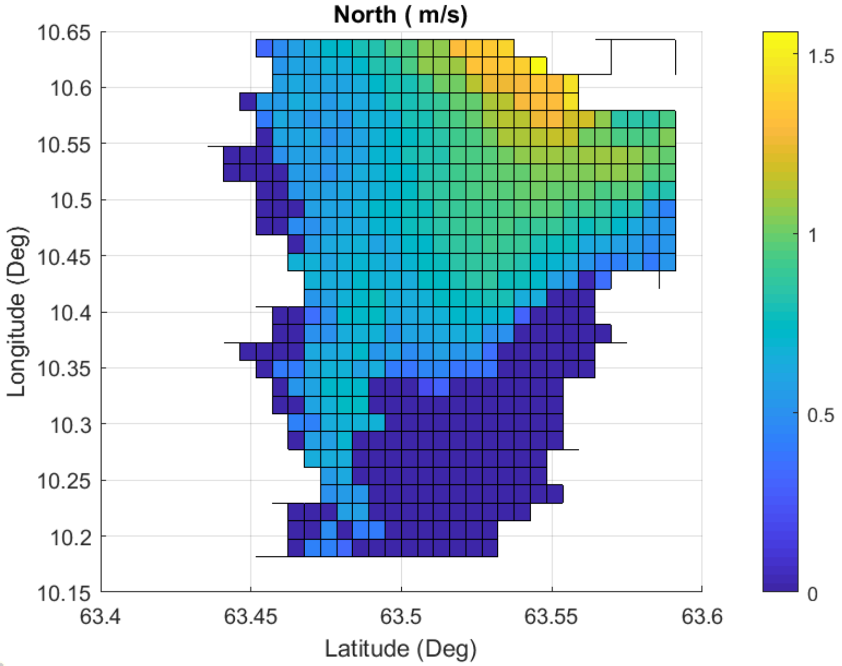


Figure 31: Figure showing the estimated northward NED velocity for each point in a grid covering the fjord just outside Trondheim city

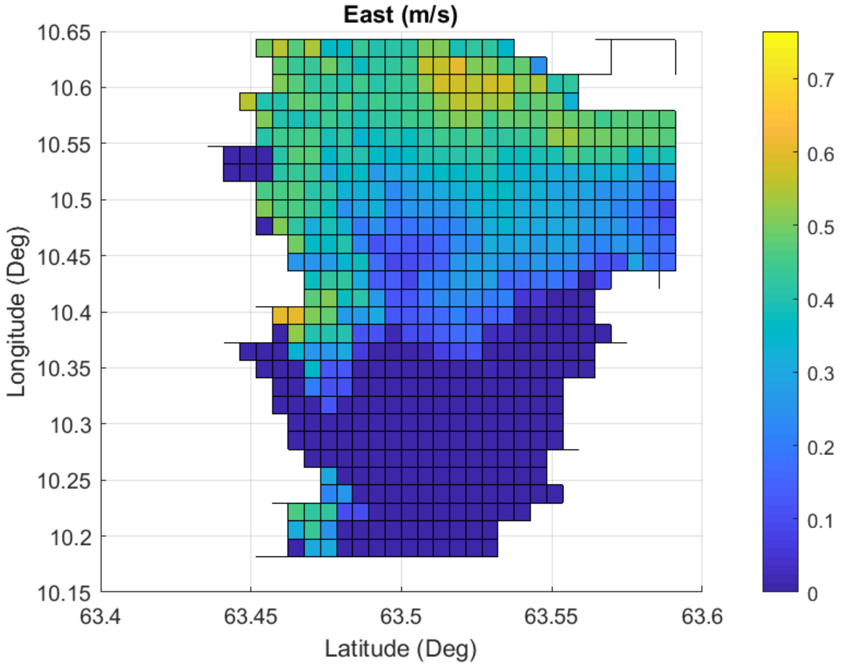


Figure 32: Figure showing the estimated eastward NED velocity for each point in a grid covering the fjord just outside Trondheim city

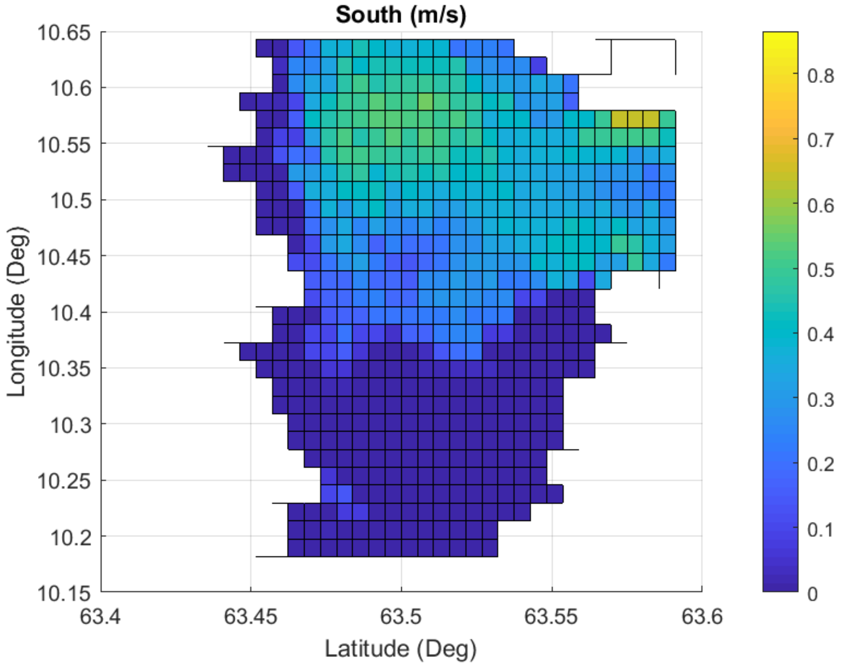


Figure 33: Figure showing the estimated southward NED velocity for each point in a grid covering the fjord just outside Trondheim city

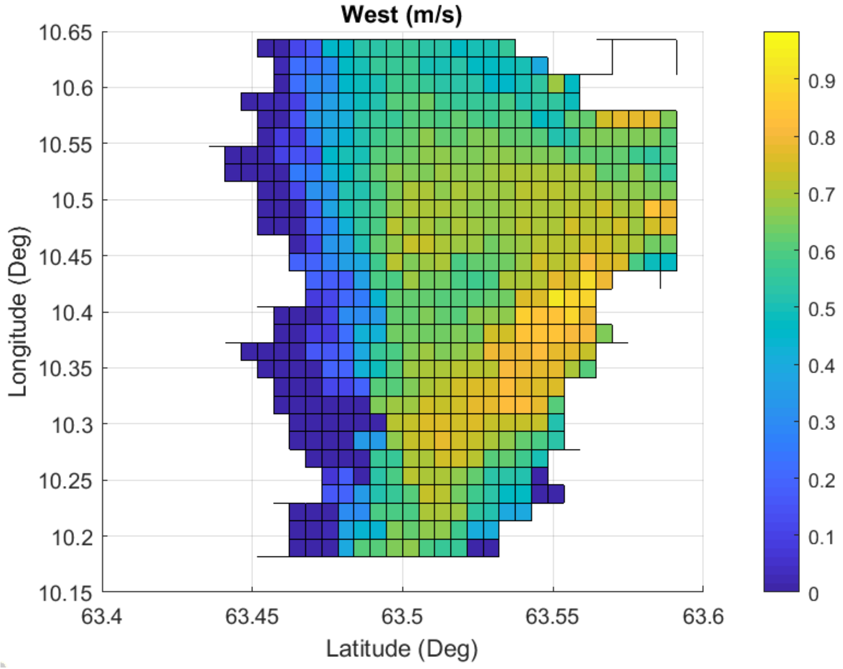


Figure 34: Figure showing the estimated westward NED velocity for each point in a grid covering the fjord just outside Trondheim city

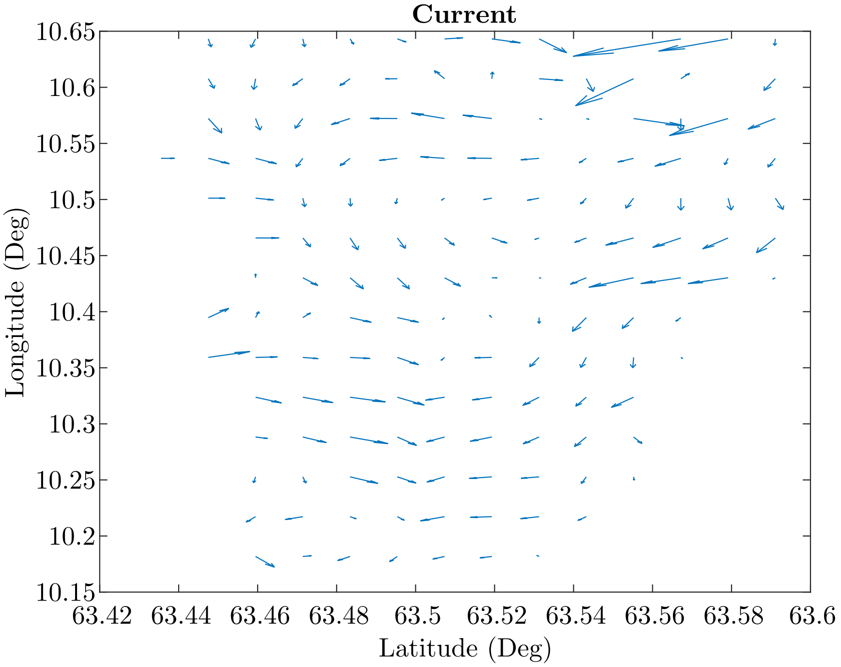


Figure 35: Figure showing current forecast data used to create feasibility map

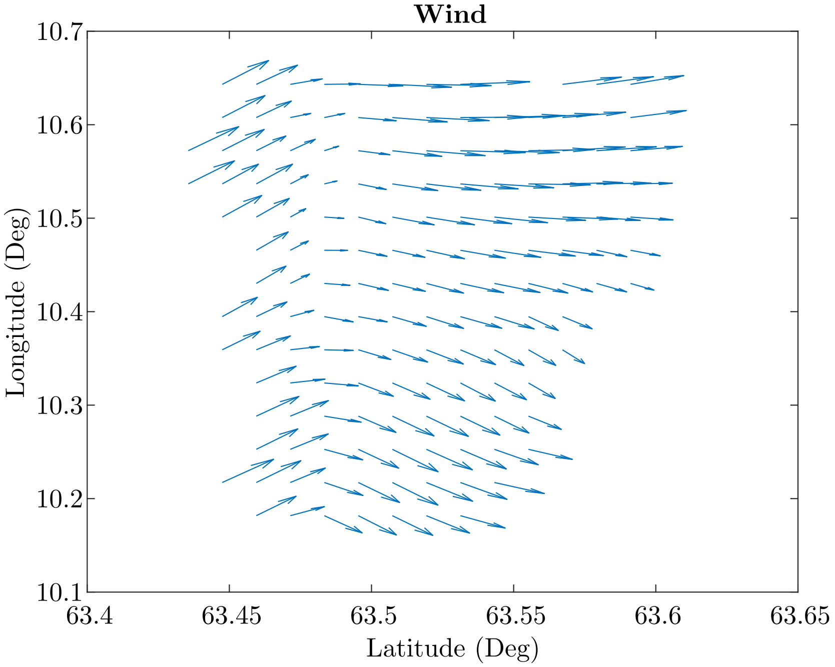


Figure 36: Figure showing wind forecast data used to create feasibility map

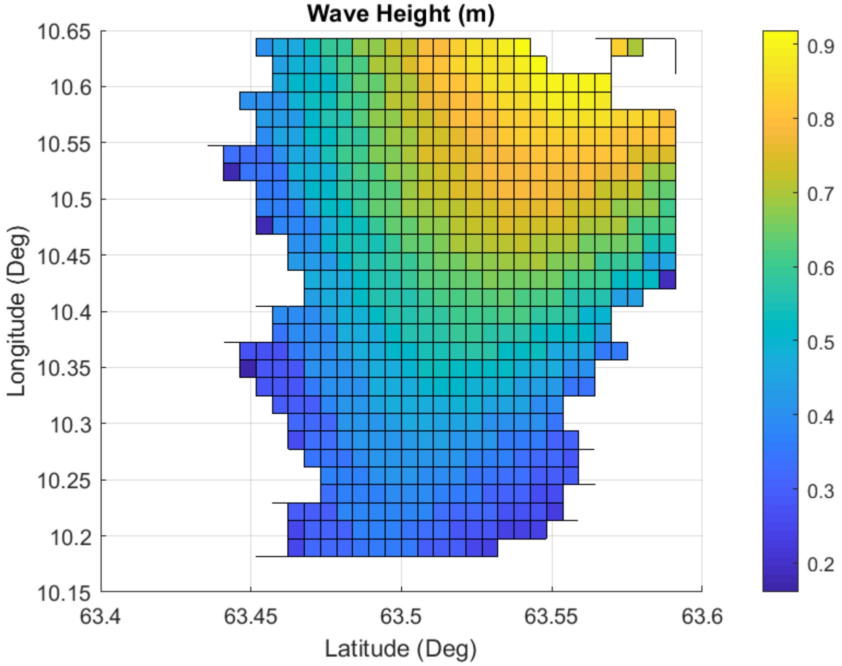


Figure 37: Figure showing current forecast data used to create feasibility map

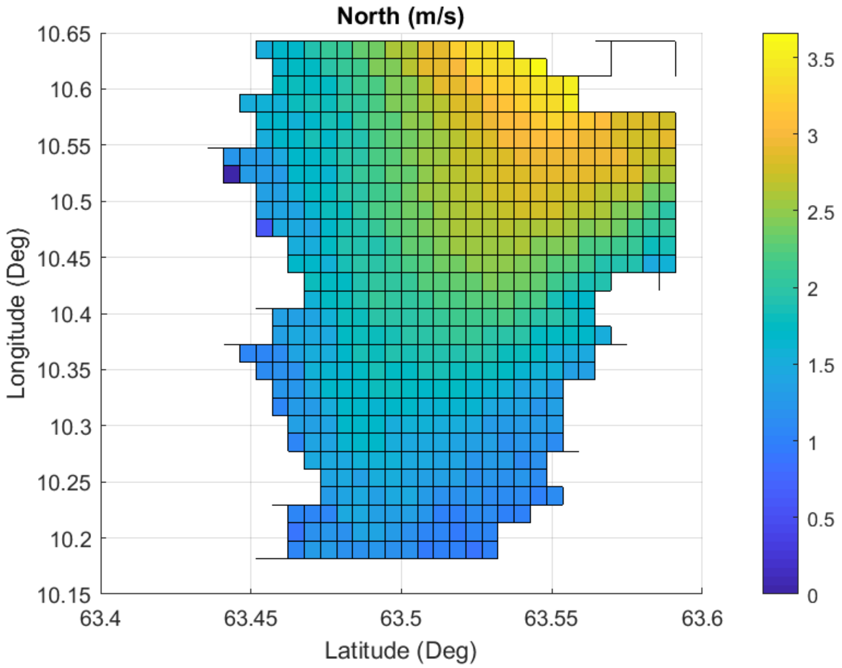


Figure 38: Figure showing the estimated northward NED velocity for each point in a grid covering the fjord just outside Trondheim city with increased hydrofoil forces

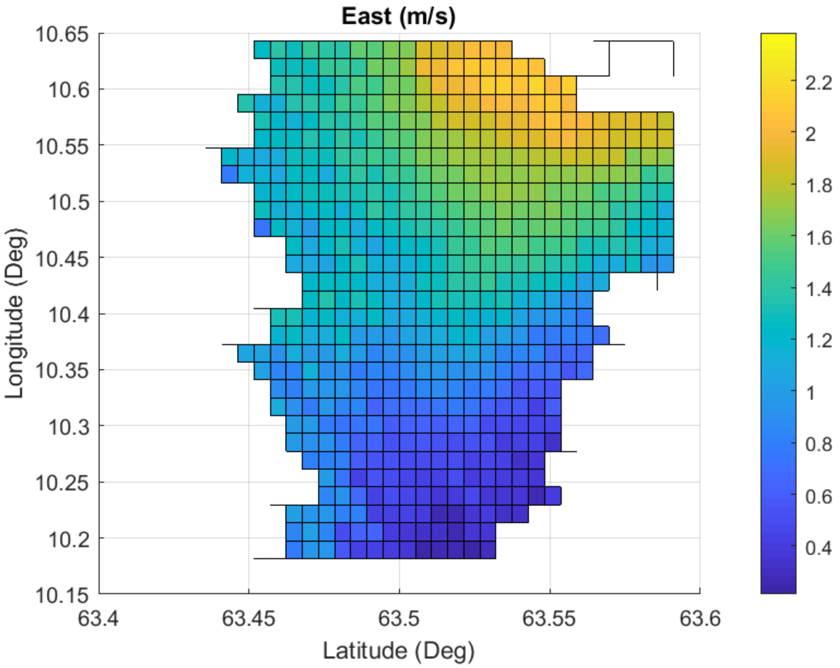


Figure 39: Figure showing the estimated eastward NED velocity for each point in a grid covering the fjord just outside Trondheim city with increased hydrofoil forces

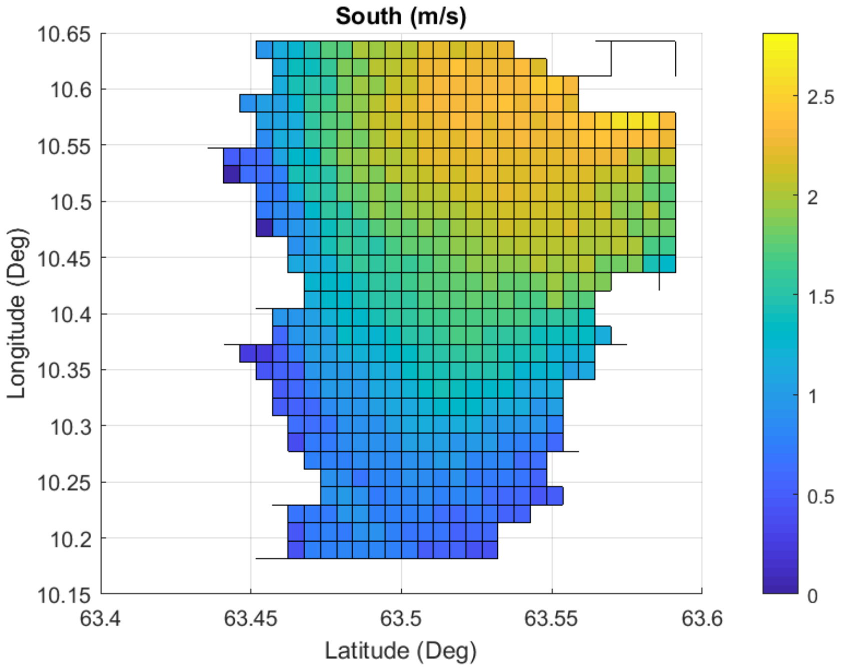


Figure 40: Figure showing the estimated southward NED velocity for each point in a grid covering the fjord just outside Trondheim city with increased hydrofoil forces

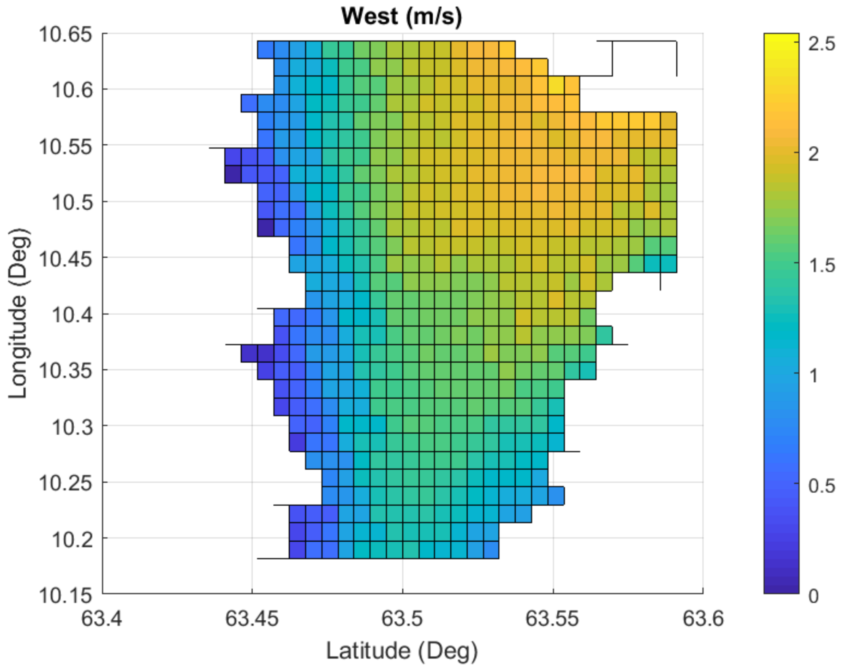


Figure 41: Figure showing the estimated westward NED velocity for each point in a grid covering the fjord just outside Trondheim city with increased hydrofoil forces

15 Summary

As stated in the start of the thesis, optimal path planning for smaller autonomous vessels allow for the autonomous vessels to more effectively use alternative propulsion methods taking advantage of weather instead of on board power to propel the vessel. To accurately describe how weather affects such vessels, an estimated model of the vessel was made which then were used in conjunction with a mission cost function to describe the cost of a mission. To be able to both optimise path and mission execution, a particle swarm was then used to optimise the cost function, allowing the optimisation algorithm to plan both vessel and sensor behaviour during a mission. Overall the structuring of the optimisation problem and optimisation showed promise for a mission planner for autonomous vessels. However, the end product was not able to find sufficient solutions given the estimated vessel parameters. A further development both in optimisation, mission cost and vessel modelling is needed to create a robust mission planner capable of guiding an autonomous vessel for months at a time.

16 Direction for Future Research

The designing of the vessel model was made using assumption for both vessel drag, hydrofoil coefficient, and weather effects on the model. Due to the limited velocities, the models was also linearised. These assumptions were made based on personal experiences. A more thorough development with a quicker feedback from tests to model modification could be beneficial. An interesting topic for future research would be to further the understanding and modelling of the hydrofoil propulsion. A more focused study on how pitch and yaw, as well as wave frequency and wave height affects vessel propulsion would greatly benefit any path planner or mission planner.

Currently, the weather forecasts are assumed to be deterministic, meaning that it is assumed that the weather forecasts are exact. One main issue is that weather forecasts today are mainly made form machine learning algorithms. Therefore, the probability distribution of the estimated parameters are not readily available. It is however possible to implement.

Another section of the thesis that can benefit from further research optimisation algorithms. The most important property of a mission planner is robustness. It is not important the vessel performs the fastest paths, but that it performs a safe and feasible path. The currently implemented particle swarm algorithm, needs manual tuning for each mission. If the tuning parameters are wrong, the algorithm will start to show undesirable behaviour, and might have difficulties finding a feasible solution. A suggestion for a more robust optimal path planner would be to combine the search capabilities of a graph search algorithm like A* or RRT* with the efficient optimisation of a particle swarm algorithm. A particle swarm could for example use a graph search to find a feasible path that could be used as a seed for the optimal path planner. It could also be interesting to test out different versions of genetic algorithms or machine learning to further the optimisation results.

17 Appendix

17.1 Use of Included Code

Together with the Thesis there is also attached the code used in the thesis. To ease the tests and use of the code for those interested, a a small tutorial for simple use of the different algorithms will be explained.

17.1.1 File Structure

The codes used can be divided into three different parts. Which is also how the code is divided. In the Master folder there is three sub folders:

- Weather
- Optimal path planning
- Parameter estimation

The different folders contain different code files and sub folders needed to achieve results relevant to the folder it is situated in.

17.1.2 Weather Folder

The weather folder contains only one matlab script, download-weather.m. This script downloads weather forecast for wind, current and waves for the area around Trondheim. The forecast horizon is predefined at the top of the matlab script. By setting startDate = [year, month, day] endDate = [year, month, day]. The forecasts can only stitch weather forecasts from end 2019 until current day. By setting correct start date and end date, the algorithm will crate an .m file with weather data between the two selected dates. Be warned, the data quickly becomes large, be therefore careful not including too many days in the forecast.

17.1.3 Optimal Path Planning Folder

N.B. For the algorithms to work, the weather forecasts have to be included. they are however half a gigabyte per day of data, so it might not be possible to upload to inspera.

The optimal path planning folder contains the optimal path planner itself PSO-testing.m. It also contains all supporting functions mission-cost.m, weatherToVelocity.m and many other algorithms to test out different parts of the algorithm. For real life tests run the PSO-testing.m file. It will run the two way-point goal mission for the date 02-20-2020. PSO-testing.m sets up the mission and runs the hybrid-PSO.m optimisation algorithm. If you want to change vessel dynamics, the different parameters are adjusted in weaterToVelocity.m file and mission cost parameters are adjusted in the mission-cost.m file.

FeasabilityMapCreator.m creates the feasibility maps and uses weaterToVelocity.m to estimate feasibility. The map limits are adjusted on line 2 and 3 where, start, end and number of intermediate points are selected. This should work for the midt-Norge area.

To test out the analytical tests, run the analytical-PSO-testing.m. . The sub folder "Analytical models" contains the analytical currents and analytical mission cost models. Analytical-PSO-testing.m does however use similar weaterToVelocity.m and hybryd pso function as the real life Optimisation algorithm, which has to be tuned to work with analytical tests.

17.1.4 Parameter estimation Folder

In the "Parameter estimation" folder the main file is the parameterEstimation.m file. If you run this file you will get the parameter estimation for the AutoNaut in the Trondheim fjord 20-02-2020. Most of the other files are mainly files, but there is also an analytical model where the least squares estimation can be compared to a test where you know the parameter results beforehand called analytical.m. By running this file you get the estimated parameter results with noise on the current and wind. It can also be observed that introducing noise to the GPS data ruins the parameter estimation. Therefore the parameter estimation in the thesis should be taken with a spoonful of salt.

References

[1] KYLE MIZOKAMI. Hugin san french vessell. https://www.popularmechanics.com/military/navy-ships/a28472065/minerve-submarine-found/. Accessed: 2020-11-17.

[2] Elaine Maslin. Hugin san juan vessel. https://www.maritimeprofessional.com/news/ocean-infinity-hunt-submarine-juan-356809. Accessed: 2020-11-17.

[3] Alberto Dallolio, Bendik Agdal, Artur Zolich, Jo Arve Alfredsen, and Tor Arne Johansen. Long-endurance green energy autonomous surface vehicle control architecture.

[4] Dr. Mark Neal. Microtransat challenge history. https://www.microtransat.org/history.php. Accessed: 2019-20-12.

[5] Claes Tretow. Design of a free-rotating wing sail for an autonomous sailboat, 2017.

[6] Saildrone homepage. https://www.saildrone.com/. Accessed: 2020-14-11.

[7] Sailbouey site. http://www.sailbuoy.no/. Accessed: 2019-23-12.

[8] Dhanushka Kularatne, Subhrajit Bhattacharya, and M Ani Hsieh. Time and energy optimal path planning in general flows. In *Robotics: Science and Systems*, 2016.

[9] Dhanushka Kularatne, Subhrajit Bhattacharya, and M Ani Hsieh. Optimal path planning in time-varying flows using adaptive discretization. *IEEE Robotics and Automation Letters*, 3(1):458–465, 2017.

[10] Enric Galceran and Marc Carreras. Efficient seabed coverage path planning for asvs and auvs. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 88–93. IEEE, 2012.

[11] Mario Arzamendia, I Espartza, DG Reina, SL Toral, and Derlis Gregor. Comparison of eulerian and hamiltonian circuits for evolutionary-based path planning of an autonomous surface vehicle for monitoring ypacarai lake. *Journal of Ambient Intelligence and Humanized Computing*, 10(4):1495–1507, 2019.

[12] Yogang Singh, Sanjay Sharma, Robert Sutton, and Daniel Hatton. Path planning of an autonomous surface vehicle based on artificial potential fields in a real time marine environment. 2017.

[13] Zheng Zeng, Karl Sammut, Lian Lian, Fangpo He, Andrew Lammas, and Youhong Tang. A comparison of optimization techniques for auv path planning in environments with ocean currents. *Robotics and Autonomous Systems*, 82:61–72, 2016.

[14] FL Pereira, J Pinto, JB Sousa, RMF Gomes, Gil Manuel Gonçalves, and PS Dias. Mission planning and specification in the neptus framework. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3220–3225. IEEE, 2006.

[15] Diogo Freitas, Luiz Guerreiro Lopes, and Fernando Morgado-Dias. Particle swarm optimisation: A historical review up to the current developments. *Entropy*, 22(3):362, 2020.

[16] Alireza Marandi, Farzaneh Afshinmanesh, Mahmoud Shahabadi, and Fariba Bahrami. Boolean particle swarm optimization and its application to the design of a dual-band dual-polarized planar antenna. In *2006 IEEE international conference on evolutionary computation*, pages 3212–3218. IEEE, 2006.

[17] Autonaut homepage. https://www.autonautusv.com/home. Accessed: 2020-14-11.

[18] Dalollio. Autonaut wiki. http://autonaut.itk.ntnu.no/doku.php?id=start. Accessed: 2019-20-12.

[19] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[20] Anna Trevisan and Luigi Palatella. Chaos and weather forecasting: the role of the unstable subspace in predictability and state estimation problems. *International Journal of Bifurcation and Chaos*, 21(12):3389–3415, 2011.

[21] BW Shen, RA Pielke Sr, X Zeng, JJ Baik, S Faghih-Naini, J Cui, R Atlas, and TA Reyes. Is weather chaotic? coexisting chaotic and non-chaotic attractors within lorenz models. In *The 13th Chaos International Conference (CHAOS2020)*, pages 9–12, 2020.

[22] Benny Dwi Kifana and Maman Abdurohman. Great circle distance methode for improving operational control system based on gps tracking system. *International Journal on Computer Science and Engineering*, 4(4):647, 2012.

[23] R Bullock. Great circle distances and bearings between two locations. *MDT, June*, 5, 2007.

[24] Forecastsnorwaycurrentwind. https://thredds.met.no/thredds/catalog/fou-hi/norkyst800m-1h/catalog.html. Accessed: 2020-11-21.

[25] Forecasts. https://www.ecmwf.int/en/forecasts. Accessed: 2020-11-21.

[26] Forecastsnorwaywave. https://thredds.met.no/thredds/catalog/fou-hi/mywavewam800mhf/catalog.html. Accessed: 2020-11-21.

[27] You-Feng Cheng, Xiao Ding, Wei Shao, and Bing-Zhong Wang. Planar wide-angle scanning phased array with pattern-reconfigurable windmill-shaped loop elements. *IEEE Transactions on Antennas and Propagation*, 65(2):932–936, 2016.

[28] Nanbo Jin and Yahya Rahmat-Samii. Hybrid real-binary particle swarm optimization (hpso) in engineering electromagnetics. *IEEE Transactions on Antennas and Propagation*, 58(12):3786–3794, 2010.

[29] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the rrt. In *2011 IEEE International Conference on Robotics and Automation*, pages 1478–1483. IEEE, 2011.

Sindre Løining Skaar

NTNU
Norwegian University of
Science and Technology