Subsea Cable Tracking using Sensor Fusion on an Autonomous Underwater Vehicle

Eirik Østeby

Marine Technology
Submission date: July 2017
Supervisor: Martin Ludvigsen, IMT
Co-supervisor: Petter Norgren, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Abstract

Surveys of underwater power cables by Remotely Operated Vehicles (ROVs) are time demanding and expensive. We want to examine the possibility of using an Autonomous Underwater Vehicle (AUV) with magnetometers for estimating the cable position and autonomously track it, given some prior information of the trajectory. Using AUVs could potentially reduce the duration of the operation significantly, reduce the personnel required, and in the future, remove the dependency on ships throughout the operation.

This thesis proposes a cable detection algorithm that utilize the side scan sonar (SSS) which is already installed on the NTNU REMUS 100 AUV. A Multi-Rate Extended Kalman Filter (MR-EKF) with a cable model is used to combine cable position estimates from magnetometers, SSS and prior information. The estimates are then used to calculate a line-of-sight (LOS) heading for cable tracking.

The results show how the cable detection algorithm can successfully find several cables with SSS data from sea trials in Trondheim Fjord, Norway. The simulations also show how the REMUS 100 can be used to track relatively thin cables with only the SSS or in combination with other sensors and a priori data using the MR-EKF. The current solution is set up for detection and tracking of a single cable and should therefore be modified if there are multiple cables in close range. The cable model in the filter enables the AUV to estimate the cable cross-track error in periods with missing measurements, if the cable heading is constant. The cable detection algorithm manages to process the SSS image in a matter of milliseconds which indicates that it can be used for online tracking although this was not tested.

Cable detection with magnetometers is already used by Statnett SF with ROVs since their submarine cables are mostly buried. However, if the cables are not completely buried the SSS solution can be very helpful. This is especially true in areas where the cables create complex magnetic fields or when twisted three-phase cables cancel out the magnetic field. Combining multiple sensors and prior information in the MR-EKF will therefore make an AUV capable of handling challenging tasks and perform the surveys more efficient than with ROVs.

Sammendrag

Inspeksjon av strømkabler på havbunnen med fjernstyrte undervannsfarkoster (ROV-er) er tidskrevende og kostbart. Vi ønsker å se på muligheten til å bruke autonome undervannsfarkoster (AUV-er) med magnetometre for å estimere dybde og posisjon til kabel, for så å følge den autonomt gitt noe forhåndsinformasjon om banen. Bruk av AUV-er kan potensielt redusere operasjonens varighet betydelig, redusere nødvendig personell, og i fremtiden fjerne avhengigheten av skip gjennom hele operasjonen.

I denne avhandlingen presenteres en kabeldeteksjonsalgoritme ved bruk av sidescan-sonaren (SSS) som allerede er installert på NTNU sin REMUS 100 AUV. Et multi-rate utvidet Kalman filter (MR-EKF) med en kabelmodell brukes til å kobinere kabelestimat fra magnetometre, SSS og forhåndsinformasjon. Estimatene brukes så til å finne en LOS-retning for å følge kabelen.

Resultatene viser hvordan kabeldeteksjonsalgoritmen korrekt klarer å finne flere kabler med SSS data fra forsøk i Trondheimsfjorden, Norge. Simuleringsresultatene viser også hvordan REMUS 100 kan brukes til å følge etter relativt tynne kabler ved hjelp av SSS, eller kan brukes i kombinasjon med andre sensorer og forhåndsinformasjon i MR-EKF. Nåværende løsning er satt opp til å detektere én enkel kabel og bør derfor modifiseres dersom det er flere kabler innen sensorenes rekkevidde. Kabelmodellen i filteret gjør det mulig for AUV-en å estimere avstanden til kabelen i perioder uten målinger, gitt at kabelen beholder samme retning. Kabeldeteksjonsalgoritmen klarer å prosessere SSS-bildet på noen millisekunder, noe som indikerer at den kan brukes i sanntid for å følge kablene selv om ikke dette er blitt testet.

Kabeldeteksjon med magnetometre blir allerede brukt av Statnett SF med ROV-er siden kablene stort sett er begravet. Hvis de ikke er helt begravd så kan SSS-løsningen være likevel være nyttig. Det er den spesielt i områder hvor kablene lager komplekse magnetfelt eller når vridde trefase-kabler kansellerer hverandres felt som vanskeliggjør magnetisk deteksjon. Å kombinere flere sensorer og forhåndsinformasjon i MR-EKF vil derfor gjøre AUV-er i stand til å håndtere utfordrende oppgaver og gjøre inspeksjonen raksere enn med ROV-er.

Preface

This Master thesis in cable tracking at NTNU serves as the final part of the study program Marine Technology. The project is carried out in cooperation with Statnett SF in the period between January and July 2017.

The core aim of the thesis is to look at the possibility of using an Autonomous Underwater Vehicle (AUV) to perform surveys of subsea power cables. Presently, this is usually done by remotely operated vehicles (ROVs).

It is assumed that the reader possess a basic knowledge within engineering science.

Trondheim, July 2, 2017

Eirik Østeby

Eirik Østeby

Acknowledgment

First, I would like to thank my supervisor Professor Martin Ludvigsen and my co-supervisor PhD Candidate Petter Norgren for facilitating the sea trials and supporting me throughout the project. The help provided during periods with failing equipment and changing of plans was crucial to the development of the project.

I would like to thank Anders Ballari and Espen Kiær from Statnett for taking their time to join us on a sea trial in Trondheim fjord this spring and for highly valuable meetings. Their initiative for this project has given me the opportunity to learn about the current state and potential of an interesting field.

Thanks to Øystein Sture for useful discussions on side scan sonar theory and practical solutions and also to Kay Arne Skarpnes for help with the logistics and navigation during the field tests.

E.Ø.

Contents

Chapter 1

Introduction

1.1 Background

Statnett SF is responsible for managing the Norwegian power grid and therefore needs to keep track of the conditions of the power cables. Underwater power line inspection is today usually performed by using ROVs, moving quite slowly, making the operation expensive. Statnett SF wants to use AUVs with magnetometers mounted on the vehicle for estimating the cable position and autonomously track the cable, given some prior information about the cable trajectory. Using AUVs could potentially reduce the duration of the operation significantly, reduce the personnel required, and in the future, remove the dependency on ships throughout the operation.

1.2 Objectives

This thesis will focus on cable detection using side scan sonars (SSS) and sensor fusion while the Master's thesis of Aksel Kjetså checks the possibility of using magnetometers. The main objectives of this Master's project are

1. Perform literature study of the state-of-the-art.

2. Write a Windows application in C++ which detects cables from SSS image.

3. Optimize cable detection algorithm with respect to speed and success rate

4. Implement a sensor fusion algorithm to combine available data in the Cable Tracker plugin

5. Field testing with the AUV REMUS 100

1.3 Limitations

Unlike some other unmanned underwater vehicles (UUVs), REMUS 100 is not specifically made for cable tracking. An AUV made for this might benefit from good maneuverability and a customized shape for the magnetometers. The first point is especially true in areas where cables lies in bumpy or steep hills which the AUV must be able to follow. The shape can be important because the measurement quality retrieved from the magnetometers depends on their spacing and distance from noise sources.

The SSS image processing is limited by the AUV hardware and the resources available on the Payload Processor in REMUS. It does not have a dedicated graphics card which would have been very helpful for processing speed. During the spring of 2017 is was found that the REMUS interface is not compliant with the installed sonar with respect to retrieving live side scan data. This means that the line detection algorithm only will be able to give the cable location after the mission or during simulations until this is fixed.

1.4 Approach

This thesis will examine the state-of-the-art cable tracking methods before looking into how this can be implemented on the AUV REMUS 100 and especially with side scan sonars. A program for detecting cables in SSS images needs to be implemented in an efficient way both regarding memory and control processor unit (CPU) usage so that tracking can be performed live. Further, the algorithm parameters should be optimized for cable tracking to avoid false negative and false positive by testing on gathered SSS data. We will first see how the program can track cables in a processed image from Sea Scan Survey before feeding it with raw data and comparing the results. If the SSS is to be used live on the AUV, the program should manage both cases. A sensor fusion filter is to be implemented in the Cable Tracker plugin so that the cable position and heading can be estimated by combining a prior, magnetometer and SSS data. The solution will then be tested on the AUV to see how well it performs.

1.5 Structure of the Report

The rest of the report is organized as follows. Chapter 2 gives an overview of the possible cable tracking methods and how sensor fusion can be applied to our problem. Chapter 3 and 4 are respectively an introduction to REMUS 100 and a description of the Cable Tracker Plugin which will be used on it. Chapter 5 explains how the SSS image processing and cable detection is implemented in a C++ program. Chapter 6 explains the different missions and simulations, followed by the results and discussion in Chapter 7. Finally, the concluding remarks are given in Chapter 8. Acronyms used throughout the thesis can be found in Appendix A.

Chapter 2

Cable Tracking

This chapter first describes relevant literature on cable tracking and how cables can be detected from SSS images by Hough Transformation. It will then show how measurements can be combined with a Multi-Rate Extended Kalman Filter and how line-of-sight (LOS) can then be using for tracking the cable.

2.1 Cable Tracking Literature

There are a few options for tracking subsea cables with different streghts in different cases. Visual tracking with camera is possible in areas where the cable is not buried, but Statnett SF buries the cables at about 1 m below the seabed to protect the cables. There are acoustic detection methods which are able to detect buried cables, but they suffer from burial depth limitations because of rapid attenuation (Szyrowski et al., 2013a). Electromagnetic detection is therefore a promising option which is they already use with ROVs.

There are different electromagnetic detection methods for finding buried subsea cables and ferromagnetic objects as described by Szyrowski et al. (2013b). One can use an active or a passive approach, and for the latter method one can also choose between using alternating current (AC) or direct current (DC). The passive approach consists of inducing a current onto a transmission line and measuring the strength of the magnetic field. This usually requires onshore control of the power line to set the desired current. In the active approach, the magnetic field is emitted by equipment on-board a vessel, which then induces a current in the target. After switching off the magnetic field, eddy currents in the target will produce a magnetic field which can be measured.

Most of the magnetic cable tracking equipment available today of the authors knowledge are meant for ROVs, vessel fixed or towing. Some of these cable tracking systems include TSS 350, Orion and Smartrak which use the passive approach. Innovatum Ltd (2012), the provider of Smartrak, ensures that the sensors also are designed for AUVs and has had a project with an AUV for pipeline and cable survey.

The KDD R&D Laboratories have however shown, with their Aqua Explorer (AE) family, successful sea trials and inspection of subsea cable tracking and measurements of burial depth by AUVs. The prototype Aqua Explorer 1000 (AE1000) was able to find and track cables already in the early 90s. AE1000 was fairly large and had a mass of 500 kg in air, making the deployment and recovery difficult. The successor Aqua Explorer 2 (AE2) was an improvement for the task, having almost half the mass of AE1000 and 20 hours longer operating time, being able to operate for 24 hours (Asakawa et al., 2002).

A recent research article from Huazhong University of Science and Technology, China, examines how an under-actuated AUV in the horizontal plane (3 degrees of freedom) with two tri-axial magnetometers can track a subsea cable using the passive approach in numerical simulations. The researchers Xiang et al. (2016) introduce a magnetic LOS guidance law, with feedback linearization, based on the trigonometry between the magnetometers and the cable, driving the horizontal offset and the course tracking error asymptotically to zero. The article shows by the simulations that the AUV is able to track a straight cable both with and without noise, but that the controller can be improved to handle unknown environmental disturbances. The authors explain the theory in a good manner and also show insight in the latest development in the field.

The proceedings by Menconi and Liparoto (1992) discuss the available methods for fault detection and tracking of undersea cables, mainly by using a surface vessel. In ideal conditions one can pin-point the exact location rapidly from onshore by using time domain reflectometry (TDR). TDR works by sending a voltage pulse through the cable and monitoring the properties of the reflected signal. However, in practice TDR is only able to give a first estimation of the fault location. The authors use the terms active and passive different than Szyrowski et al. (2013b) and Xiang et al. (2016), differentiating active and passive tracking by *if* current is induced in the target object instead of *how*. Thus, by passive tracking they mean detecting disturbances to the magnetic field caused by a metallic object while by active tracking they mean detecting a cable with current. Two active tracking methods are discussed, namely the use of electrodes to detect the electric field or by using a coil to detect the magnetic field. They suggest using the first method in the beginning from longer distances and switching to a magnetic probe when the vessel is getting close.

The author finds the amount of scientific papers on online cable tracking by SSS limited. The proceedings of Petillot et al. (2002) show how pipelines can be detected with a SSS by line fitting before manually driving the vehicle above the cable and then change to detection by multi-beam echo sounder. The large size of subsea pipelines makes it difficult know how relevant the results are for smaller power cables. However, it is interesting how a priori information of pipeline dimensions and maximum curvature are used to constrain the cable model. The proceedings by Bagnitsky et al. (2011) simulates online cable tracking with an AUV by creating SSS images with a seabed model. They calculate the image gradients and find lines by a modified Hough transform where variables are calculated relative to a geographical coordinate system.

The discussed papers and proceedings show that tracking of underwater cables with AUVs is still a relevant topic with the very recent paper by Xiang et al. (2016), even though solutions were made already in the early 90s with the AE family. Menconi and Liparoto (1992) gives a good introduction by explaining the different methods for finding a fault in a subsea cable. Hagen and Kristensen (2002) show how easy and generic the Hugin payload system is, which is also used by Remus 100, thus showing good potential for adding plugins and magnetometers. Although there are solutions to the discussed topic already, it would seem like this project could contribute to the field as the overall research available is somewhat limited and that the project seems feasible with REMUS 100.

2.2 Cable Detection Using Side Scan Sonar

This section describes the process of finding a cable in a SSS image by using edge detection and Hough Transform.

2.2.1 Image Filtering

The image need to be filtered to lower the effect of noise on the gradients which are higly sensitive to this. Gaussian filters are often used in image processing because they remove the high frequency components and thus act like low-pass filters. However, Bagnitsky et al. (2011) change each pixel to the nearest value of the surrounding pixels in a sliding 3x3 window, arguing that the filter requires low resources and will not change pixel value on thin cable-like borders.

2.2.2 Gradients

Image gradients are useful for detecting cables since large intensity variations can be expected. Different types of gradient operators can be used by convolution to estimate the image gradients. The Sobel operator is shown in (2.1) and (2.2) as 3x3 matrices where the horizontal and vertical gradients are estimated. The total gradients of the image **I** can then be approximated by (2.3).

$$\mathbf{G}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \mathbf{I} \tag{2.1}$$

$$\mathbf{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \mathbf{I} \tag{2.2}$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \tag{2.3}$$

Other types of operators can also be used, like the Scharr or Prewit operator or the second order Laplace operator. Bagnitsky et al. (2011) use the Scharr operator to calculate the gradients and divide them by the length per pixel in each direction. It is reasonable to normalize the gradients with respect to the length per pixel because these depend on the sensor speed v, slant range R (one way distance of SSS signal) and AUV altitude. Assuming a flat seabed, the distance to a point in the SSS image in each direction can be found as follows:

$$dx = dx_{ss} \pm \sqrt{R^2 - (h_{AUV} + h_{ss})^2} \tag{2.4}$$

$$dy \approx 2Tv \tag{2.5}$$

where dx_{ss} and h_{ss} are the horizontal and vertical placement of the SSS relative to the AUV, h_{AUV} is the AUV altitude and T is the one way travel time of the signal. dx is positive for starboard and negative for port side. The image scale can also be corrected by using (2.4) and (2.5) so that the relative cable heading can easily be found. The gradient values that passes a lower threshold are called edge points and can be used for detecting lines with the Hough Transform.

2.2.3 Canny Edge Detector

The Canny edge detector is a popular algorithm for finding edges in images and was developed by John F. Canny in 1986, although it can be used with an arbitrary number of dimensions. There are three performance criteria for the algorithm as described in the article by Canny (1986): Good detection, good localization and only one response to a single edge. The first criterion means that edge points should have a high probability of being correctly detected and the second means that the center of the edges found should be close to the center of the true edges.

The Canny edge detector works by first applying a Gaussian filter and finding the intensity gradients of the image. Non-maximum suppression is then applied to the image, based on the gradients, to remove pixels that are not a part of an edge. The final step is to apply an upper and a lower threshold using hysteresis. All edges with an intensity gradient above the upper threshold are accepted and all edges below the lower threshold are discarded. Edges with a value between the upper and lower threshold are included if they are connected to a part above the upper threshold. In the implementation of Canny, the ratio between the upper and lower threshold is in the range of 2:1 and 3:1. The result is a binary image of the edges.

2.2.4 Hough Transform

This section will describe the describe the Hough Transform and some alternative versions of it. Note that even though we will focus on line detection, the Hough Transform can be in general be used to detect any arbitrary shape as shown by Ballard (1981).

Standard Hough Transform

The Standard Hough Transform (SHT) is a method used in computer vision for finding lines, curves, circles, ellipses, etc. in an edge image. The geometrical figure is found by mapping an image from the Cartesian coordinate space, e.g. x and y, to the n-dimensional Hough space, where n is the number of parameters needed to represent the shape. Only two parameters are needed to represent a straight line in polar coordinates (ρ,θ). The perpendicular distance ρ from origin to the line is allowed to be negative and so the angle θ must have a range of 180° to cover exactly all possibilities. All edge points will be given a weight for every discretized value of θ since they can potentially be a part of a line in any direction its transform

A pseudo code for finding lines with the SHT is shown in Algorithm 1. The perpendicular distance is found by Operation 2 for each edge pixel and for a discrete number of angles. The resulting lines given by $(\lfloor\rho\rceil,\theta)$, where ρ is rounded, obtain weights which are added to an accumulator matrix **A**. Points on a straight line will accumulate weights for all possible directions, but even more in the direction and distance representing the line in the image. Thus, lines can be defined by setting a minimum threshold for the accumulated weights and discarding the rest.

```
1 Initialize A to 0 ;
  for Each (x,y) pixel do
      if pixel == edge pixel then
          for $\theta = 0$ to $\pi$ do
2             $\rho = x * \cos(\theta) + y * \sin(\theta)$;
3             Increase $\mathbf{A}(\theta, \lfloor\rho\rceil)$ by 1;
```

Algorithm 1: SHT for detecting lines in a 2-dimensional Cartesian space

A single step in the procedure is illustrated for an edge pixel in Figure 2.1 with only four discrete angles. Since the maximum length of ρ will be half of the diagonal ($\frac{1}{2}d$ pixels), the accumulator will be a d x 4 matrix since ρ also can be negative (as with ρ_4). The accumulator

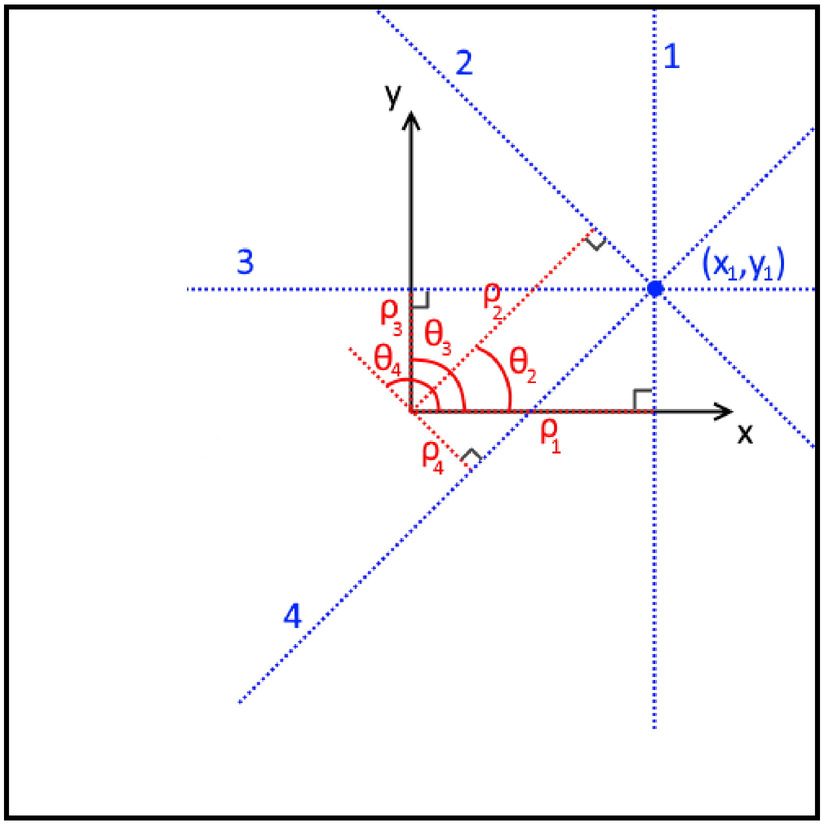


Figure 2.1: Hough transform illustration of an edge point from an image

value for the pairs (ρ_i,θ_i) will be incremented equally by a value of one for this step. If there were to be another edge point at line number 4, then the value corresponding to (ρ_4,θ_4) would be 2 and the rest would be lower.

A disadvantage with the SHT is the computational power needed to do the calculations for every pixel. Also, SHT is a one-to-many mapping since a single edge point gives a line (or surface depending on n) in the Hough space, meaning that a larger amount of memory is needed. Thus, a more efficient method may be necessary if the image processing is to be performed live on the AUV.

Alternative Hough Transforms

There are several alternative Hough Transforms which tries to solve the bottlenecks of the SHT. The Randomized Hough Transform (RHT) by Xu et al. (1990) is a many-to-one mapping that selects a random sample of n edge points and transforms them into one point in the parameter space. The computation speed is greatly enhanced by only updating one parameter point for each step, especially for higher number of parameters n where the SHT would have to accumulate all the cells lying on a hypersurface. The RHT also has the advantage of an inherently high resolution and a low memory requirement because it does not discretize the parameter space like the SHT. However, Kälviäinen et al. (1995) states that complex and noisy images may be a problem for the basic RHT.

A slightly newer version named the Progressive Probabilistic Hough Transform (PPHT) also uses random sampling, but is a one-to-many mapping like the SHT. The PPHT tries to detect lines efficiently by "exploiting the difference in the fraction of votes needed to reliably detect lines with different numbers of supporting points" (Matas et al., 1998). This is possible because a short line requires a higher fraction supporting points to vote before a line is reliably detected. Unlike other probabilistic approaches this makes the PPHT able to detect lines without a priori knowledge of the number of points belonging to the line. Besides the need to do less operations than the SHT, the PPHT can be stopped at any time and still give useful results.

2.3 Multisensor Data Fusion

This section will first present the states to be estimated with a simple model for the cable. It will then show how the model can be used in an Extended Kalman Filter (EKF) and explain the multi-rate EKF (MR-EKF). The EKF will not be explained thoroughly as a great deal of literature already exists on this topic.

2.3.1 Cable Model

The cross-track error ϵ and the cable heading ψ_{cab} are proposed as the state variables $\mathbf{x}$ for the multisensor data fusion because they can be used for LOS guidance. Using the state variables, the model simply assumes that the cable continues in a straight line from the current estimated position and heading. This gives us

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\epsilon} \\ \dot{\psi}_{cab} \end{bmatrix} = \begin{bmatrix} U \sin(\psi_{cab} - \psi_{AUV}) \\ 0 \end{bmatrix} = \mathbf{f}(\mathbf{x}) \tag{2.6}$$

where $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$ and U is the forward velocity (surge). Bold font is used to represent vectors and matrices as opposed to scalars. By adding system noise **w** and measurement noise **v**, the nonlinear system becomes:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{E}\mathbf{w} \tag{2.7}$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{2.8}$$

where **y** is the measured states. The gradient of **f**(**x**) shown in (2.9), the Jacobian, will be useful for the next section.

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 & U\cos(\psi_{cab} - \psi_{AUV}) \\ 0 & 0 \end{bmatrix} \tag{2.9}$$

2.3.2 Extended Kalman Filter

A discrete EKF is introduced to estimate the states by linearization at each time step.

$$\mathbf{x}(k+1) = \Phi(k)\mathbf{x}(k) + \Gamma(k)\mathbf{w}(k) \tag{2.10}$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \tag{2.11}$$

where k represents the time step. The matrices Φ and Γ are approximated by forward Euler integration with the sampling time h:

$$\Phi(k) \approx \mathbf{I} + h\frac{\partial \mathbf{f}(\mathbf{x}(k))}{\partial \mathbf{x}(k)}\Big|_{\mathbf{x}(k)=\hat{\mathbf{x}}(k)} \tag{2.12}$$

$$\Gamma(k) \approx h\mathbf{E} \tag{2.13}$$

The matrices can then be inserted into a discrete-time EKF (Fossen, 2011, 298):

$$\mathbf{K}(k) = \bar{\mathbf{P}}(k)\mathbf{H}^T[\mathbf{H}\bar{\mathbf{P}}(k)\mathbf{H}^T + \mathbf{R}]^{-1} \tag{2.14}$$

$$\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}\bar{\mathbf{x}}(k)] \tag{2.15}$$

$$\hat{\mathbf{P}}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}]\bar{\mathbf{P}}(k)[\mathbf{I} - \mathbf{K}(k)\mathbf{H}]^T + \mathbf{K}(k)\mathbf{R}\mathbf{K}^T(k) \tag{2.16}$$

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k) + h\mathbf{f}(\hat{\mathbf{x}}(k)) \tag{2.17}$$

$$\bar{\mathbf{P}}(k+1) = \Phi(k)\hat{\mathbf{P}}(k)\Phi^T(k) + \Gamma(k)\mathbf{Q}\Gamma^T(k) \tag{2.18}$$

where (2.14-2.16) represent the update cycle and (2.17-2.18) represent the prediction cycle. The current state estimate $\hat{\mathbf{x}}(k)$ is found by using the predicted value from (2.17) and correcting it by multiplying the Kalman gain $\mathbf{K}$ with the residual $\mathbf{r}(k) = \mathbf{y}(k) - \mathbf{H}\bar{\mathbf{x}}(k)$. The Kalman gain used here is called the optimal Kalman gain because it gives minimum mean square error (MMSE) estimates of $\mathbf{x}(k)$. The process- and measurement noise covariance matrices Q and R will be tuned since we do not have any prior information of the covariance.

The **E** matrix will be set as in (2.19) assuming that there is no correlation between the process noise on the two states. The $(states * sensors)$ x $states$ observation matrix **H** will only have 1's and 0's since the states are obtained directly from the samples. Equation (2.20) is used when combining two sensors, and **H** will look similar in other cases.

$$\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.19}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.20}$$

2.3.3 Multi-rate Extended Kalman Filter

A MR-EKF handles the problem of fusing measurements from multiple sensors with different sampling rates. It has the advantage of allowing the prediction and update step in the filter to be performed at a high frequency while the sensors sample at their own frequency. This means that new information can be utilized quickly instead of waiting for the slowest sensor. As explained by Armesto et al. (2007), it uses multi-rate holds (MR-holds) and multi-rate samplers (MR-Samplers) to apply the input **u** and output **y** at the prediction and update step respectively. The idea is to generate high frequency samples from lower frequency inputs and outputs.

The first step of the MR-EKF is to apply the MR-holds to the previous inputs and MR-sampler on the current measurements. The MR-holds are not implemented here because there is currently no input in our filter. The MR-Sampler will only include measurements from sensors with new samples during the last time period, giving a varying sized measurement vector **y**. This means that samples from none, some or all the sensors may be included. The MR-EKF then performs the conventional prediction equations (2.17-2.18) but disregard the update cycle (2.14-2.16) if no new samples were made.

We want to combine a priori, magnetometer and SSS estimates of the states by using the MR-EKF. The processed estimate samples are regarded as sensor measurements of the cable cross-track error and heading although they are not sensor output in the strict definition. This may cause some unusual filtering behavior because the samples have already been processed and since measurements might not always be provided regularly.

2.4 Line-Of-Sight

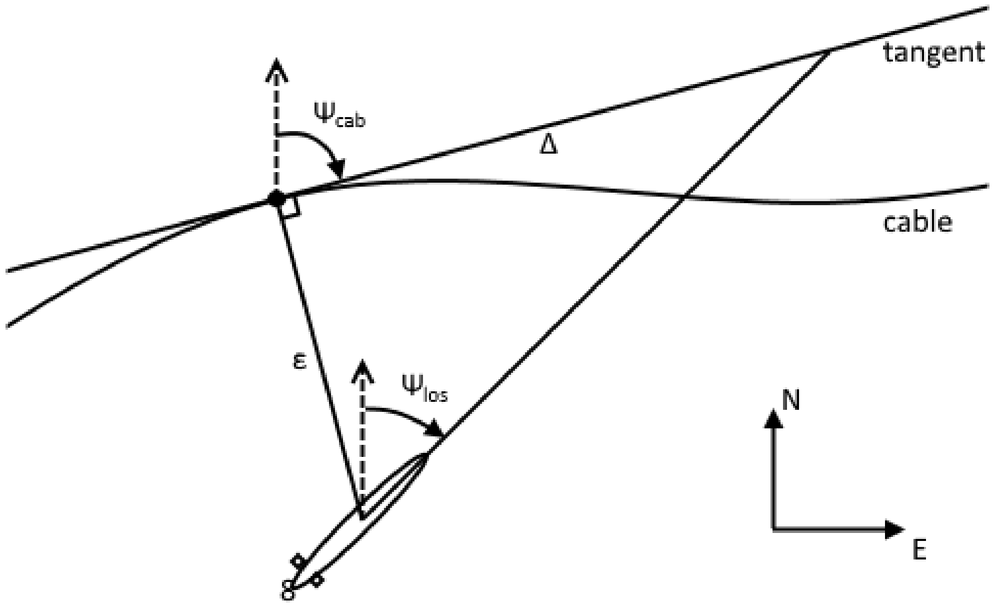


Figure 2.2: LOS illustration

With the state estimates from the previous section one can use LOS to make the vehicle intercept the cable as illustrated in Figure 2.2. The desired LOS heading can be calculated as a function of the cross-track error ϵ.

$$\psi_{los} = \psi_{cab} - \arctan\left(\frac{\epsilon}{\Delta}\right) \tag{2.21}$$

where $\Delta > 0$ is the lookahead distance which is how far along the cable tangent the vehicle should aim. The moving point on the cable is taken as the nearest point like in the filter. The lookahead distance can be chosen freely larger than zero. A small value will lead to an aggressive control law and vice versa. Since this is practically a proportional (P) controller it will not be able to handle a constant disturbance.

As described by (Fossen, 2011, 262), it is also possible to add an integral term to enable a vehicle to follow a straight line even when affected by currents and nonzero sideslip angles. The steering law then becomes

$$\psi_{los} = \psi_{cab} - \arctan\left(K_p\epsilon + K_i\int_0^t \epsilon(\tau)\mathrm{d}\tau\right) \tag{2.22}$$

where $K_p = \frac{1}{\Delta} > 0$ and K_i tunable control gains. In this case, care must be taken to prevent integral windup and overshooting of the cable track.

Another possible method to handle disturbances is to estimate the side-slip angle and subtract this from (2.21) as done by Xiang et al. (2016). The side-slip angle can be found as $\beta = \arctan\frac{v}{u}$, where u and v are the surge and sway velocities respectfully defined in the body frame. Having no integral, this method avoids the windup problem.

Chapter 3

REMUS 100

This chapter will introduce the AUV REMUS 100, its simulator and the HUGIN AUV Payload system.

3.1 The Vehicle

The REMUS 100 shown in Figure 3.1 on the next page is a NTNU owned AUV from the Kongsberg company Hydroid. It has a depth rating of 100 m, a maximum speed of 4 knots and a typical endurance of 10 hours at this speed (Hydroid, 2010). It can run predefined missions, but some simple commands can also be sent to the acoustic modem on the AUV. The commands are sent by using a handheld device called a Digital Ranger connected to a submerged transducer called tow-fish. During the mission, the vehicle will transmit modem messages once every 50 s. These messages will be processed by the Ranger and transferred to the vehicle interface program (VIP) on a topside computer.

3.2 Navigation System

3.2.1 Long Baseline

Long baseline (LBL) is a positioning method often used for large area survey missions. It uses triangulation to find the position of the AUV by interogating two or more transponders placed on

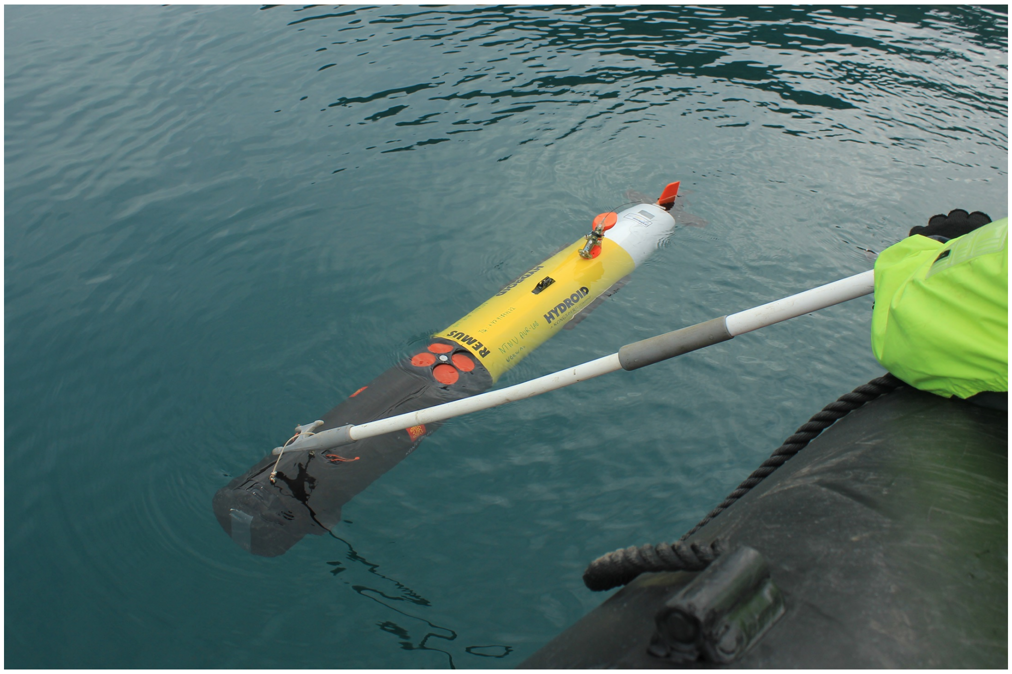


Figure 3.1: The REMUS 100 deployed at Svalbard in August 2016

the seabed using sound waves. By measuring the round-trip travel time, from sending a signal from the AUV to receiving a response from the transponder, one can calculate the distance in between by using an estimate of the sound velocity. Using LBL, the position is updated every few seconds. REMUS then use dead reckoning between each received signal. The Maximum depth rating for the transponders is 30 m and the range is 2 km (Hydroid, 2010).

3.2.2 Dead Reckoning

Dead Reckoning is a way of estimating the position based on speed, heading and previous positions. The speed is estimated by using the ADCP explained below and the propeller RPM. The heading is estimated by using the vehicle's compass and accelerometers/gyros. The error in the estimated position and orientation will only increase as the speed and accelerations are integrated over time. Thus, a regularly position fix with for instance GPS or LBL is necessary to limit the errors.

3.2.3 Acoustic Doppler Current Profiler

An acoustic doppler current profiler (ADCP) can be used to find the relative velocity between the AUV and its surroundings or to obtain a velocity profile of the water column. This is possible due to the Doppler frequency effect caused by the relative displacement between the signal source and the target (Ludvigsen, 2016). The ADCP module can be seen in Figure 3.1 as the part with four red transducers pointing upwards. It also has four transducers pointing downwards. When achieving bottom lock, REMUS can find the speed relative to the ground and use this for dead reckoning. REMUS needs to be within 30-40 m from the sea bottom to obtain bottom lock (Hydroid, 2010).

3.2.4 GPS

GPS radio signals cannot reach the AUV when submerged, but it can be used to occasionally obtain a position fix when transponders are unavailable. One can either include a GPS position fix at the surface in the mission or the REMUS will do this automatically if the error is found to be large and some other conditions are met.

3.2.5 Side Scan Sonar

The REMUS 100 has a 900 kHz Marine Sonics Technology Sea Scan module installed with two transducers mounted on the starboard and port side. A SSS sends out sound waves and records backscattered signals and environmental noise. The image created by a SSS is based on the travel time T of the signal. The traveled distance, called the slant range R, can be found as (3.1) with an estimate of the sound velocity.

$$R = \frac{cT}{2} \tag{3.1}$$

The maximum range of the SSS can be set in the REMUS mission file and will affect the cross-track distance of the seabed that the sonar covers, namely the swath width. The beam angles are automatically adjusted during the trip to control the range by assuming a constant sound velocity of 1500 m/s.

3.2.6 Magnetometer Housing

An extension of the AUV called the magnetometer housing will be made and fitted with two magnetometers and a Raspberry PI. The Magnetometers will be placed in waterproof containers on the outside of the AUV to get more spacing between the two. The plan is to connect them to a Raspberry PI inside the AUV extension. The housing is a part of the wet section, meaning that it will take in water. Thus, the Raspberry PI must also have a waterproof container. Kjetså is managing the Raspberry PI that will process the magnetometer readings and send cable position estimation to the Cable Tracker Plugin described in the next chapter.

3.3 HUGIN AUV Payload System

The conference proceedings by Hagen and Kristensen (2002) describe how the software system of HUGIN AUV is built to easily support all kinds of new and different payloads, making the AUV flexible for new applications. The HUGIN Software System is expained because of the similarities to the system in REMUS 100. See Figure 3.2 for a block diagram of the latter.

3.3.1 HUGIN Software System

Normally the AUVs will have three main computers which are communicating over a network using Common Object Request Broker Architecture (CORBA): The Payload Processor (PP), the Navigation Processor (NavP) and the Control Processor (CP). CP represents the nerve center of the AUV, while NavP is optional and PP can be replaced if desired. The idea behind the “plug and play” payload system is that one only has to create a plugin for each new payload installed on the AUV, without changing any of the hardware or software previously installed. The only requirement is that the plugin has some minimum pre-defined functions, for instance telling the plugin if the payload is powered or not. One can then either use the basic graphical user interface (GUI) given or create a more advanced one to handle the plugin from the topside. The proceedings work as a good introduction for those who want to develop a payload for a HUGIN or REMUS AUV that has this system.

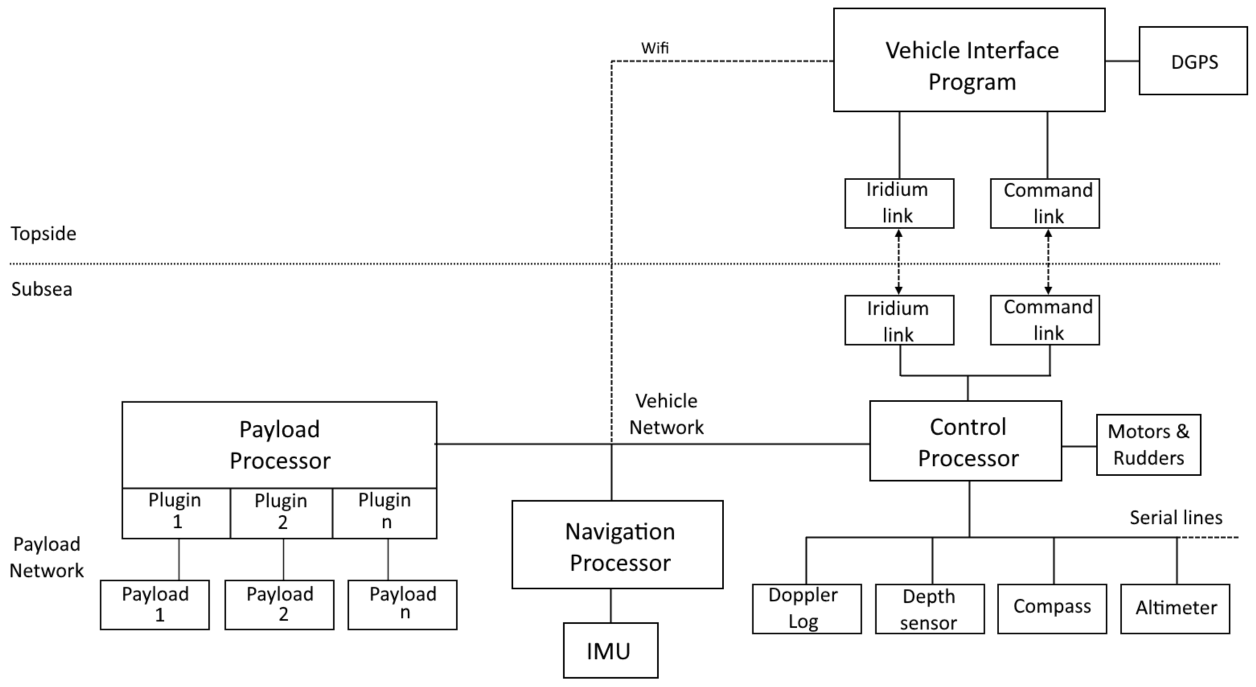


Figure 3.2: Block diagram of Remus 100. An altered version of the HUGIN block diagram of Hagen and Kristensen (2002)

.

3.3.2 REMUS 100 Components

Figure 3.2 shows the main components of the REMUS 100 in use. The vehicle works much in the same way as the HUGIN Software System, but it does not have all of the same links between the AUV and topside. The iridium link is a satellite communication system used to locate the vehicle when out of range or if the system fails. The WiFi connection to the vehicle network is only available when the AUV is in close range and not submerged. The payloads have their own network between PP and the payloads. In this way, a plugin cannot for any reason overload the vehicle network because they are separated.

3.3.3 HUGIN 1000 Payload SDK

Hugin 1000 Payload SDK is a software development kit from Kongsberg Maritime AS initially made for the AUV Hugin 1000, but it is also used with the Remus 100. The plugin can therefore be used in both. The SDK contains components like library files and development tools necessary for developing software for the AUVs.

3.4 REMUS Simulator

The REMUS simulator in Figure 3.3 works so that one can connect to and interact similarly as with the real AUV. The simulator runs the operating system VxWorks and is connected to a Windows computer with VIP using an Ethernet cable. The settings for the simulated AUV can be set in the vehicle configuration file as usual. Some of the special parameters used for simulation includes a speed up factor, speed/direction of current and initial depth/position of vehicle.

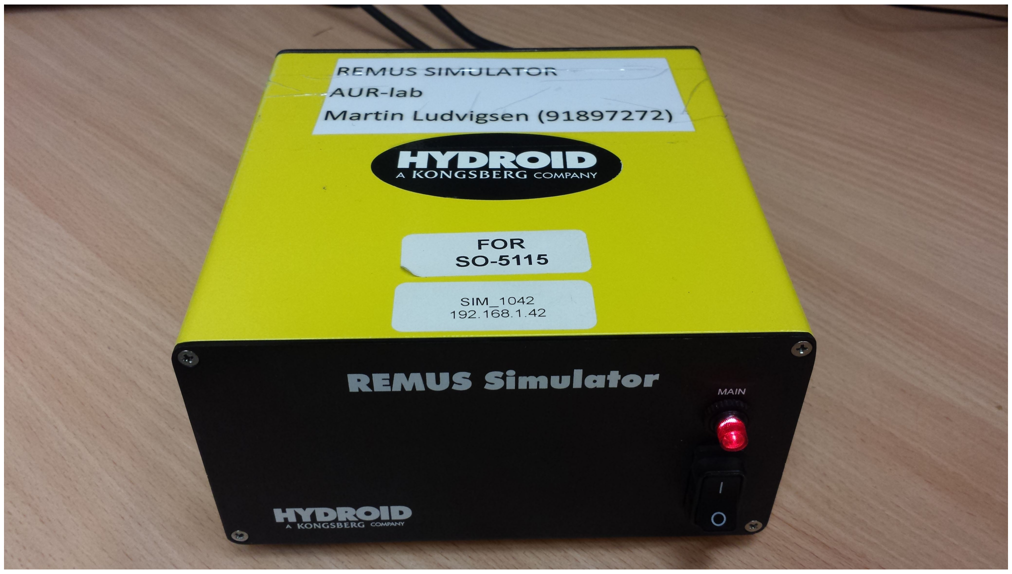


Figure 3.3: REMUS simulator

Chapter 4

Cable Tracker Plugin

This chapter describes the Cable Tracker Plugin developed for use on the REMUS 100 AUV and more specifically the PP.

4.1 Code structure

Figure 4.1 is a structure chart showing some of the basic parts of the Cable Tracker Plugin, where GuidancImpl was mostly written during the autumn of 2016. An instance of RemusReconImpl is created when the plugin is started. This also leads to the creation of the other objects with associated functions, as illustrated, and their main purposes are explained below. GuidanceImpl will also create a CtdThread, which is not shown, that can be used to obtain measurements of conductivity, temperature and depth (CTD). New children of RemusReconImpl include the SideScanThread, NavThread and MagnetometerThread used for transmitting data, SensorFusionThread for combining measurements and the DebugThread used for testing and simulations. As the names suggests, the threads run simultaneously by using multithreading.

4.1.1 RemusReconImpl

As a parent of the other C++ objects, RemusReconImpl takes care of the required supervisory functions. PowerOn needs to be called to inform the plugin that its payload power has been turned on, while activate will start the RECON driver and all of the threads. One can also do commands, turn on or off logging of sensor data and more. Figure 4.2 on page 25 shows a test

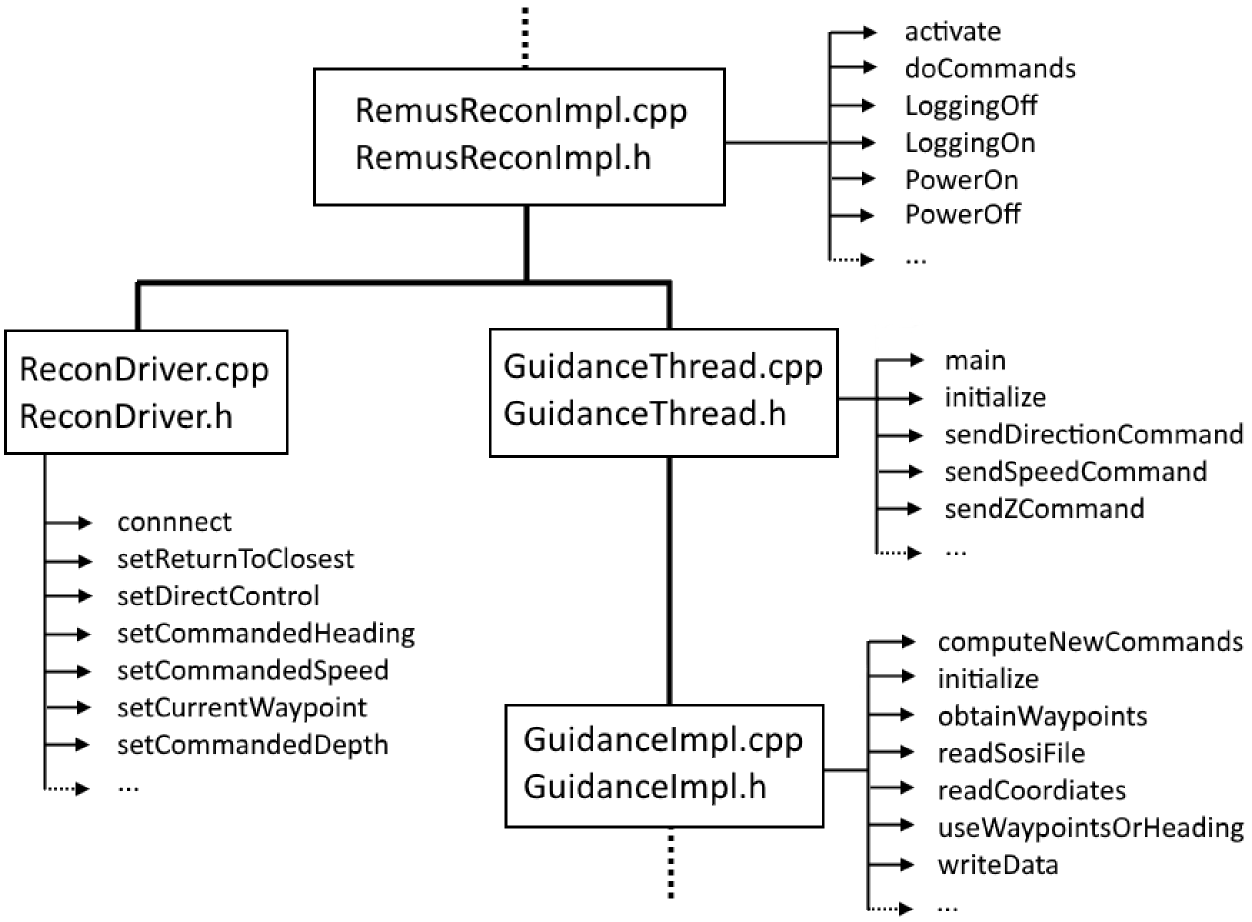
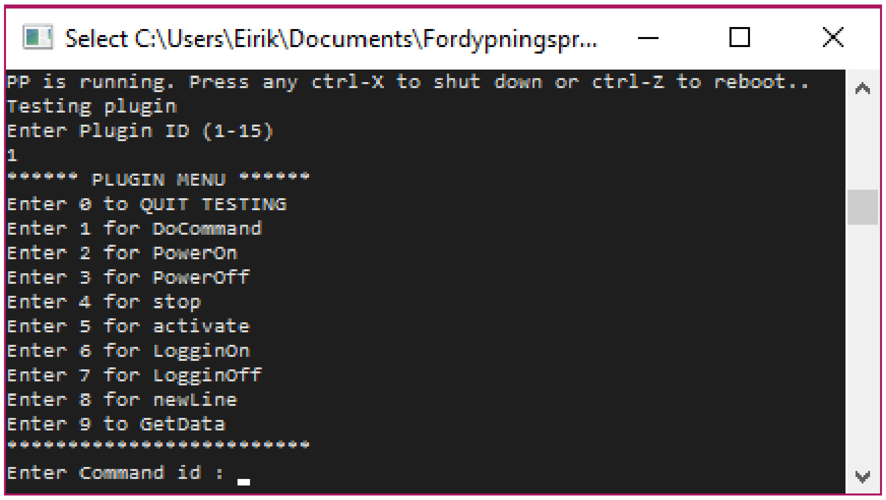


Figure 4.1: Structure chart showing the basic parts of the plugin

run of the Cable Tracker Plugin on PP where one can use commands that will be handled by RemusReconImpl.

4.1.2 ReconDriver

The RECON (Remote Control) driver takes care of the communication between the plugin and the REMUS vehicle. In this way, the plugin can obtain state information and send navigation and control commands. setReturnToClosest was added to make the AUV return to the closest point on track after the plugin disables guidance override.



```
PP is running. Press any ctrl-X to shut down or ctrl-Z to reboot..
Testing plugin
Enter Plugin ID (1-15)
1
****** PLUGIN MENU ******
Enter 0 to QUIT TESTING
Enter 1 for DoCommand
Enter 2 for PowerOn
Enter 3 for PowerOff
Enter 4 for stop
Enter 5 for activate
Enter 6 for LogginOn
Enter 7 for LogginOff
Enter 8 for newLine
Enter 9 to GetData
*************************
Enter Command id : _
```

Figure 4.2: PP.exe test with Cable Tracker Plugin

4.1.3 GuidanceThread

The Guidance Thread uses the initialization (INI) file CableTracker described in Section 4.2 to decide when to take control of the AUV with RECON and what type control should be used. It will call GuidanceImpl to compute new commands and send them by using the Recon Driver.

4.1.4 GuidanceImpl

The C++ constructor of GuidanceImpl starts out by creating and starting a CTD thread before running initialize. Depending on the configuration in the INI file, it will obtain and transform waypoints from a given SOSI file with readSosiFile or from a data (DAT) file with readCoordinates. SOSI ("Samordnet Opplegg for Stedfestet Informasjon") is a standarized method for sharing georeferenced data made by the Norwegian Mapping Authority, namely Statens Kartverk. In case of a SOSI file, the plugin will utilize the info in the file to transform any type of coordinates into latitude/longitude in radians. If the plugin is provided with a DAT file, it expects two space separated columns of coordinates in radians. useWaypointsOrHeading was created as a decision logic to enable the plugin to switch between cable tracking with measurements and predefined waypoints. Currently it simply chooses waypoints until the RECON driver is active to prevent strange behavior.

A priori Guidance

This function uses predefined waypoints to determine the LOS parameters based on the current position. The coordinates are transformed from latitude and longitude coordinates to cross-track error ϵ and cable heading ψ_{cab} by finding the nearest waypoints and using them in the right order per the planned direction. A simple arcus tangens of the coordinate differences was used at first as an approximation of the cable heading. This was found to be very inaccurate for the intermediate directions northeast, southeast, southwest and northwest because of the spherical shape of the earth. Instead the bearing angle must be used which gives the required heading for going from one point to another. The latitude α and longitude β for each point are inserted into (4.1) and (4.2) to map the coordinated to a flat surface where X and Y represent movement east and north. The desired cable heading can then be calculated by (4.3) where $atan2$ is used to get the correct value.

$$X = \cos(\alpha_2) * \sin(\beta_1 - \beta_2); \tag{4.1}$$

$$Y = \cos(\alpha_1) * \sin(\alpha_2) - \sin(\alpha_1) * \cos(\alpha_2) * \cos(\beta_1 - \beta_2) \tag{4.2}$$

$$\psi_{cab} = \arctan\frac{Y}{X} - \frac{\pi}{2} \tag{4.3}$$

4.1.5 SideScanThread, NavThread

The SideScanThread and NavThread transmit SSS and navigation data respectively to the Cable Detector application and receive information of detected cables. The information is transmitted with Winsock, where the plugin acts as the server and the application as the client. Each thread waits for new data to be written to the database. When this happens, the thread will read the data and send it to the client. The threads accept incoming UDP messages from any address and therefore need a message from the client to know where to send the data. For simplicity, the two threads use separate sockets and ports although they could have used the same socket.

4.1.6 MagnetometerThread

The MagnetometerThread communicates in the same way as the previous two threads, but receives data from a Raberry PI which is connected to two magnetometers. The data includes the amplitude and phase of the measured magnetic field in surge, heave and sway. An estimation of the cross-track error and the relative heading to the cable are obtained for use in the MR-EKF and also the cable depth.

4.1.7 SensorFusionThread

The SensorFusionThread aplies the theory from Section 2.3 to combine measurements from the magnetometers and SSS with a priori data. The thread waits for the specified filter time step before sampling any new measurements and performing a step of the EKF or the MR-EKF. The specified time is actually how long the thread will sleep between each step and the time step h used in the filter is therefore measured. The pseudo-random number sampling by Box and Muller (1958) is used to generate Gaussian noise in simulations.

4.1.8 DebugThread

The DebugThread is used for simulations and testing. By providing the plugin with data from missions, this thread will pass on the navigation and side scan data as if they were real measurements. This enables us to combine the measurements using sensor fusion despite the incompatibility of the REMUS interface with the sonar.

4.2 Cable Tracker Configuration File

The Cable Tracker Plugin has a configuration file (INI file) to ease the change of code parameters. In this way, the parameters can be changed in the file without rebuilding the plugin in Visual Studio (VS). An example is given in Appendix B.1. Most of the parameters will be explained here, but more details can be found in the actual file.

4.2.1 Communication

The first parameters under Communication are used by the RECON driver. Here one can change the IP and port number of the AUV/simulator.

4.2.2 Control

The parameters under Control is used by the GuidanceThread. Here one can decide when RECON is allowed to take control (StartLeg/StopLeg), what kind of control is wanted and if RECON should be allowed in case of vehicle errors. One can for instance choose between manual control of the rudders or heading control. The calculated LOS heading can simply be set with heading control by sending the desired heading.

4.2.3 Guidance

The parameters under Guidance are used by GuidanceImpl and provides information of where to find potential waypoints and which ones to use. In case a SOSI file is used, it is also necessary to provide the curve number representing the cable and if the location is in the northern (1) or southern (0) hemisphere. A ReturnToClosest value set to one will trigger the corresponding function in ReconDriver explained in Section 4.1.2. DirectionCommandAsWaypoints tells the plugin to send commands as waypoints (1), heading (2) or both (0). Some LOS parameters can also be set like lookahead distance, integral gain and saturation limit for the integral.

4.2.4 Sensor Fusion

Under Sensor Fusion one can choose between EKF or MR-EKF and set the filter time step. Which sensors to include must be specified and a folder path to where the plugin should look for the matrices E, H, Q and R as space separated text files. The initial state x_0 can also be give in a text file or it will be set to zero.

4.2.5 Cable Detector and Magnetormeter

The parameters under Cable Detector and Magnetometer are simply for setting the port number for communication with the Cable Detector and the Raspberry PI.

Chapter 5

Cable Detector

A 32 bit windows application was created to perform image processing of the SSS data by using the OpenCV (Open Computer Vision) library. The application was named Cable Detector since its task is to detect and locate cables from the images. To obtain the SSS data online the data is received from the Cable Tracker Plugin by using Winsock with the User Datagram Protocol (UDP).

5.1 Implementation of OpenCV

The reason for having another program performing the image processing is to utilize the benefits of fast image processing by OpenCV. The newer versions of OpenCV were found difficult to implement in a plugin using the old VS 2005. An advantage of having a separate program is also that the application can either run on the same computer as the plugin, namely PP, or on separate computer if preferable. To make the application able to run correctly on PP, a VS Redistributable corresponding to the VS used for building the application needs to be installed with the correct libraries. VS Redistributable 2013 was chosen because the 2015 version failed to be installed correctly on PP. The working solution was found to be a 32 bit version of OpenCV 3.0.0 compiled with CMake for VS 2013. The application can therefore run on any Windows computer with VS Rediddtributable 2013 and a single small dynamic link library (dll) file from OpenCV called opencv_world300.

5.2 Program Structure

This section will introduce each part of the Cable Detector program: CableDetector, NavThread, SideScanThread and ImageProcessor. The SideScanThread and NavThread are not to be confused with their counterparts in the plugin with the same names although they are similar.

5.2.1 CableDetector

The *main* function of the executable file lies in the part called CableDetector. It creates an object of CableDetector which will contain objects with the other threads. CableDetector will start by looking for a configuration file as described in Section 5.3 before starting the rest of the program. The threads run simultaneously by using Microsoft Windows native multithreading capabilities. The program will only stop when a user exit the program since the main function is set to wait for threads with infinite loops to finish.

5.2.2 SideScanThread

The SideScanThread was created for the purpose of receiving and storing SSS data from the plugin, but it will also send some data back. When activated, the thread will try to initialize a WinSock client with a configuration preferably matching the *plugin* SideScanThread. When running, the thread will try to receive data using a non-blocking call. This means that the thread will simply continue if no UDP message is incoming. Data from the messages received are stored chronologically in a text file. Thus, out of order messages may need to be sorted later by using the timestamp or ping number. Also, messages may be lost because of the unreliability of the UDP. The data sent back contain the estimated cross-track error and heading of lines in the SSS image found by the ImageProcessor.

5.2.3 NavThread

The NavThread works similarly as the SideScanThread except it receives navigation data instead and does not send any information back.

5.2.4 ImageProcessor

The ImageProcessor thread is where the cable is actually detected from the SSS image. It will run the LineDetection function regularly as new data arrives, but can be limited to prevent cpu overload as image processing is a demanding task.

Transmission Loss

The acoustic waves will spread out and therefore become weaker as they travel. Water will also attenuate energy from the signal between the side scan sonar and the ground. The combination of these effects are called transmission loss (TL) and will reduce the intensity of the image in regions with larger travel time or range and make it more difficult to detect cables. The two-way transmission loss can be found as (5.1) in dB by assuming a spherical sound propagation and a small reference distance as shown in Hovem (2012).

$$2TL(R) = 40\log(R) + 2\alpha R \tag{5.1}$$

where α is an absorption coefficient in dB/m and the slant range R (one way distance) for each pixel is given by $\frac{c}{2}T$. The absorption is significant for a SSS with 900 kHz sound waves. This will for instance give a coefficient of about 0.55 dB/m in water with one atmosphere pressure, 10 °C, 7.8 pH and a salinity of 35 g/kg. The transmission loss can be compensated for by adding the estimate to the backscatter strength as a time varying gain (TVG).

We do however only have an intensity value scaled between 0 and 32767 (16-bit) and will therefore just even out the image by using the proposed TVG in (5.2) and tune a manually. Also, the travel time for each point is unknown, but the range can be approximated with (5.3) by assuming a flat seabed and equal spacing between each point.

$$G(R) = aR \tag{5.2}$$

$$R(s) = R_{max}\frac{s}{S-1} \quad \forall s \in \{0, 1, .., S-1\} \tag{5.3}$$

where s is the element number and S is the total number of elements in the ping channel. The maximum slant range is set in the mission file for REMUS 100 where a constant sound velocity of 1500 m/s is assumed. A more accurate range is therefore calculated for the TVG by using the sound velocity from the CTD thread of the Cable Trakcer Plugin.

It is possible to account for changes in backcatter intensity caused by changes in sensor altitude as shown in Capus et al. (2008). The authors of the article also use data from a REMUS vehicle with the same sonar. The range and angular dependent factors needs to be treated separately under varying altitudes, so they use correction factors for compensation. We will however command a constant altitude and only correct for the transmission loss.

LineDetection

The ImageProcessor reads the SSS data text file and finds lines in the image by using functions based on the theory in Section 2.2. First, edges are found in the SSS image by e.g. the Canny edge detector with the OpenCV functions GaussianBlur and Canny. The first function is used to apply a Gaussian filter with the with a 3x3 filter kernel. The second function will output a binary image with edges based on the threshold specified in the configuration file. Other filtering and gradient/edge detector methods are also available. The next step is to perform some sort of a Hough transform of the edge image. OpenCV has efficient functions for the SHT and the PPHT discussed in Section 2.2.4, but they will not provide the weights obtained. The weights could be used as a measure of intensity in the KF or used to choose the strongest line. A problem with the HT functions from OpenCV is that they might give many lines and we only want to follow a single line. If many lines are detected, it is possible to increase the threshold and run the HT again, but this requires a lot of computational power. An alternative version of the OpenCV SHT was therefore made.

HoughTransform

The self implemented Hough Transform uses a similar coordinate system shown in 2.1 on page 11, but with θ increasing in the clockwise direction. The function will iterate through the edge image and check if there is an edge pixel at each point. If there is one, it will calculate the polar representation of every possible line. This means that the length ρ will be calculated for every discretized angle θ. The position in the Hough accumulator matrix corresponding to the rounded values will then receive one point each time they appear.

The strongest line is found as the maximum accumulation in the Hough accumulator. If the value is higher than a certain threshold, the line is assumed to be a cable and can be used for LOS guidance. The relative angle ψ_{rel} between the AUV and the cable is given by θ for $\theta < \frac{\pi}{2}$ when assuming that the AUV is not going the opposite direction and the image is scaled. The cross-track error ϵ in pixels can be found by shifting the origin to the top of the SSS image where the AUV is positioned, as in (5.5). For larger values of θ, the signs of ψ_{rel} and ϵ_{pix} are switched to avoid turning around.

$$\psi_{rel} = \begin{cases} \theta & \text{if } \theta \leq \frac{\pi}{2} \\ -\theta & \text{if } \theta > \frac{\pi}{2} \end{cases} \tag{5.4}$$

where $\psi_{rel} = \psi_{cab} - \psi_{AUV}$.

$$\epsilon_{pix} = \begin{cases} x_0 * \cos\theta + (y_0 - y_{origin}) * \sin\theta & \text{if } \theta \leq \frac{\pi}{2} \\ -(x_0 * \cos\theta + (y_0 - y_{origin}) * \sin\theta) & \text{if } \theta > \frac{\pi}{2} \end{cases} \tag{5.5}$$

where $x_0 = \rho * \cos\theta$ and $y_0 = \rho * \sin\theta$ are the orthogonal points on the line relative to the image center.

SendResult

Before sending the results, the cross-track error needs to be converted into meters and the relative heading into absolute cable heading. This was done with rough approximations, assuming constant heading and altitude, and no pitch, roll or sway in the duration of capturing the image. The AUV movement and rotation will affect the results and can be taken into account by using navigation information and the rotation matrix. To do so, the navigation data at each ping needs to be available for the Cable Detector.

5.3 Cable Detector Configuration File

The Cable Detector configuration file contains the necessary parameters to set up communication with the plugin and for optimizing the line detection algorithm. An example is shown in Section B.2. It can be used for changing the method used for estimating image gradients, finding edges and tuning. It also includes PauseMillisec and SkipArrays which can be used to reduce the computational load on PP. The first parameter pauses the thread for a given number of milliseconds between each processed image and the latter parameter makes the thread wait for a specified number of pings before proceeding. The parameter $AccumulatorReduction$ was introduced to lower the weights by $sin(\theta)$ times the factor in the SHT. This is useful if the SSS image width is much larger than the height.

5.4 Data Management

5.4.1 User Datagram Protocol

The Cable Tracker plugin communicates with the Cable Detector and the Raspberry PI by using User Datagram Protocol (UDP) packets. UDP as described in Postel (1980) enables IP communication between applications with a minimum of protocol mechanism. The protocol does not require any handshake between applications which means that it is possible to broadcast messages on a network. The minimalism of UDP makes it preferable for time-sensitive applications, although there is no guaranty that the packets arrives chronologically or at all as opposed to the Transmission Control Protocol (TCP). Having time stamps or message numbers in the message is therefore an essential measure to keep track of the message flow.

5.4.2 Memory Management

The setup between the plugin and the program allows an arbitrary size the SSS ping array. The program receives information of the array length and allocate necessary memory on the heap. Apart from the possibility of insufficient memory, the only limitation on the array length is the predefined maximum buffer size when receiving UDP data.

5.4.3 Code Efficiency

Effort is made into making the cable detection efficient because of the limited computational power of the PP computer. The sine and cosine values are pre-computed and stored as lookup tables for better performance as they are used $edges * angles$ times in the SHT. Lowering the memory needed for the accumulator by switching from unsigned integer (4 byte) to short unsigned integer (2 byte) also gave a performance boost in the offline testing. Changing to unsigned character (1 byte) did however not show any difference and would limit the maximum values in the accumulator to 255. Note that the size of C++ types varies between systems.

Chapter 6

Field Tests and Simulations

This chapter explains the setup of the field tests and simulations, and will also show some of the navigation results. The cable detection and tracking results are treated in the next chapter.

6.1 Munkholmen Power Cable

A SOSI file with information of power cables between Korsvika and Munkholmen was provided by Trønder Energi Nett AS. This information will be called *a priori* data since it is preexisting knowledge of the cable position obtained before the field test. The cable we are looking for is a TERE 1 kV power cable descibed in Draka Norsk Kabel AS (2010) which may have an outer diameter between 32 and 71 mm.

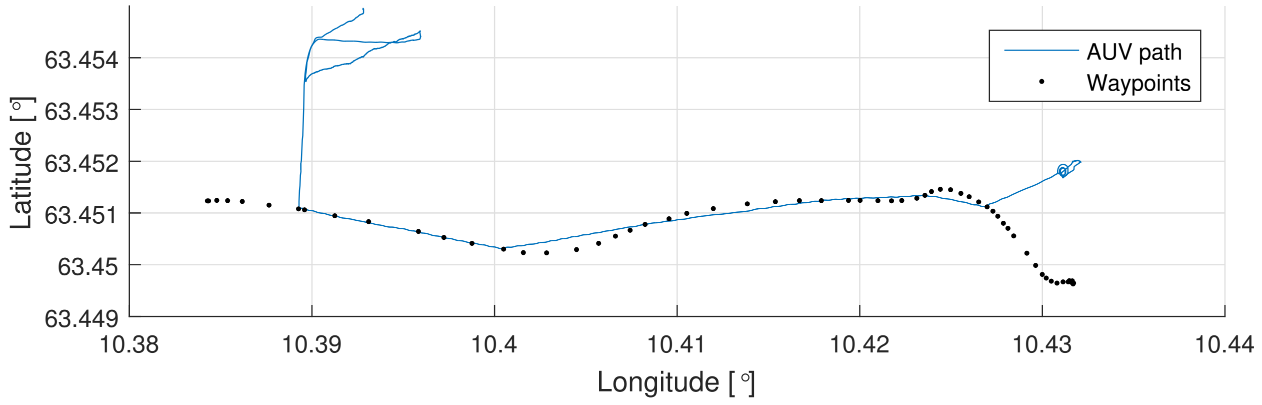


Figure 6.1: AUV position during field test March 24, 2017

6.1.1 Field Test March 24

Setup

In the field test of March 24, 2017, waypoints in a rmf mission file was used to guide the AUV eastwards from Munkholmen to Korsvika in Trondheim fjord, Norway. The LBLs were not placed since bottom lock was expected over the relatively short distance of about 2 km. The magnetometers were not installed yet and the plugin was not used. The SSS range was set to 50 m to cover a large area but still detect relatively thin cables. The speed was set to 3 knots and the altitude to 7 m.

Results

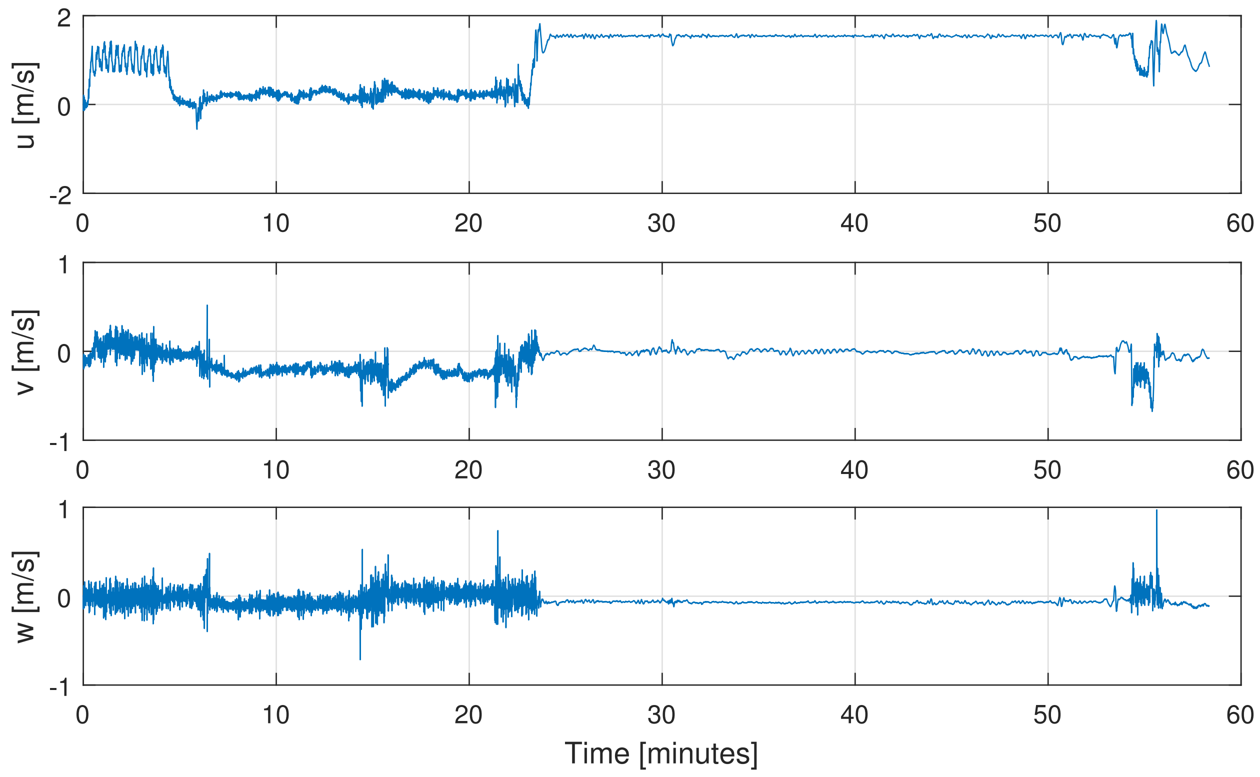


Figure 6.2: Velocity in surge, sway and heave respectively

The AUV was struggling to dive in following waves in the beginning of the sea trial and was therefore picked up added some weight. The start can be seen in the top left corner of Figure 6.1, including the part in the boat. Figure 6.4 on page 42 shows that the AUV dives after about 24 minutes and keeps an altitude of about 6-8 m after reaching the seabed. The body frame velocity in surge, sway and heave can be seen in Figure 6.2. One can see that the surge speed does not reach the setpoint of 1.5 m/s until the AUV manages to dive. The sway velocity shows that there is little side slip. The roll, pitch and yaw angles are shown in Figure 6.3. The pitch angle during the dive is mostly less than 10° in absolute value and often less than 5°, and the roll has an offset of minus 2-3°.

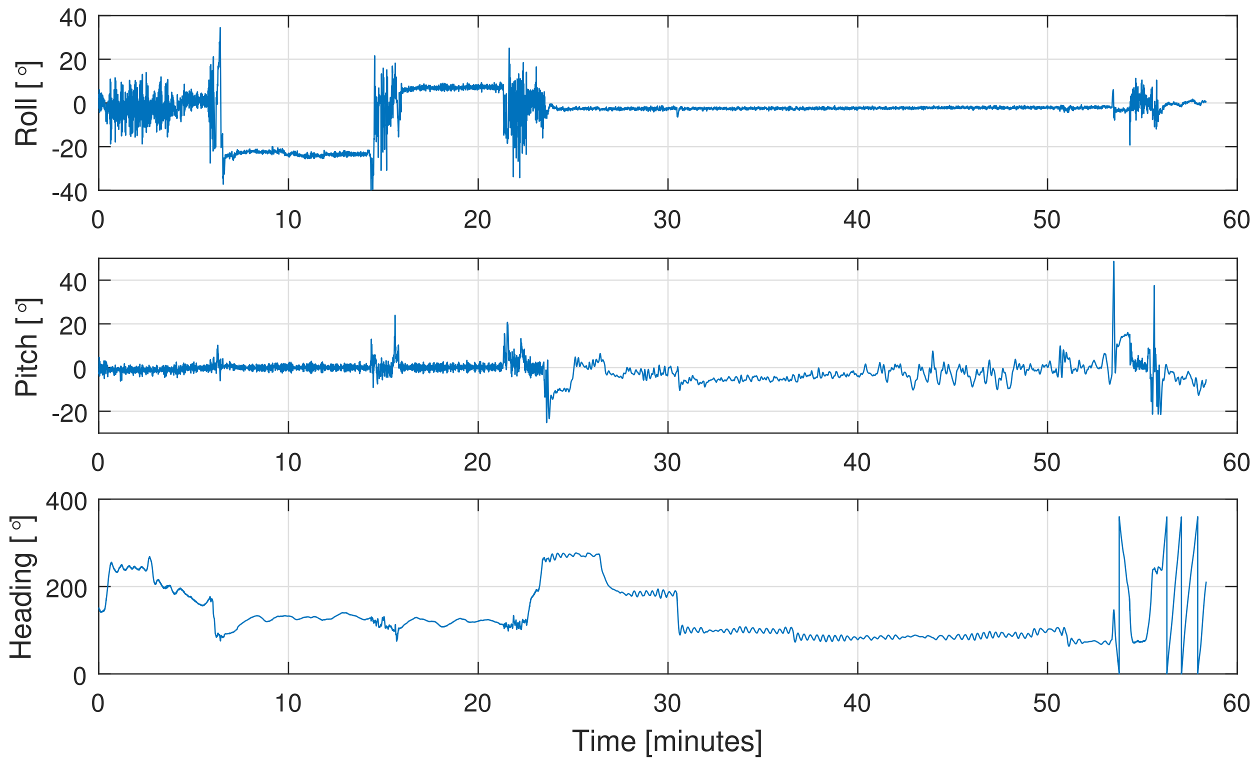


Figure 6.3: Roll, pitch and heading during sea trial March 24, 2017

Discussion

The AUV was able to get bottom lock during the whole dive which results in good positioning estimates. The heading is steady during the dive except for two sharp turns in the beginning and some oscillations. This means that small segments of SSS images can be composed with good results by combining pings without regards to the yaw rate. The large heading variations at the end are caused by the loiter objective where the AUV drives in circles. The small roll angle is caused by the thruster momentum and will vary with the RPM. The AUV manages to regulate

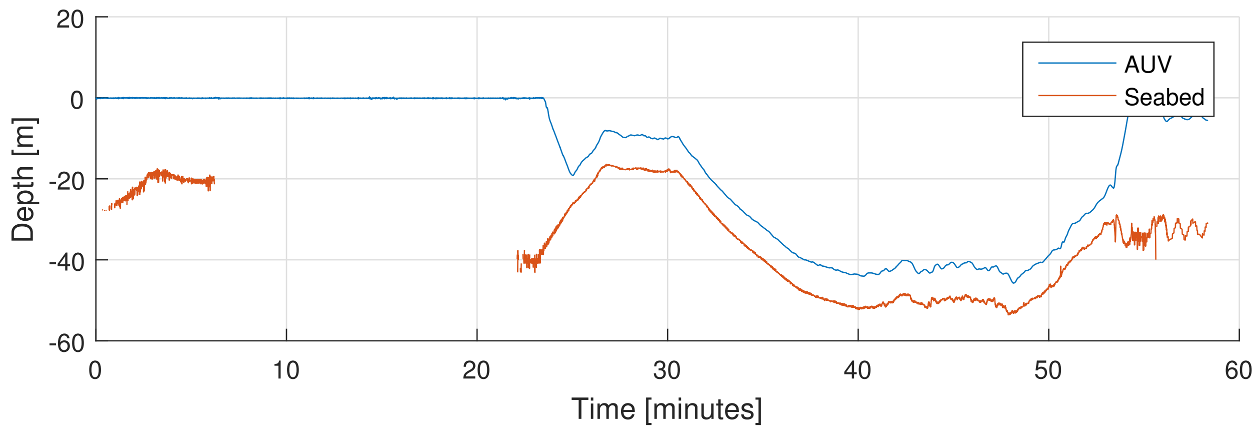


Figure 6.4: Depth of AUV and sea bottom

the altitude with 1 m margin, even when ascending, which means that it can be lowered to provide better magnetometer readings in the future when the housing is installed.

6.1.2 Field Test March 29

Setup

The setup of March 29, 2017, was similar to the previous test, but this time the plugin was used to control the AUV by RECON. A delay in RECON override was implemented to prevent a prematurely override. The speed was set to 1.5 m/s and the altitude lowered to 5 m. The plugin was configured to have an offset of 1 m north for every meter altitude to get the cable inside the SSS image.

Results

The Cable Tracker Plugin followed the waypoints but the mission was aborted because of an RECON override timeout which was not changed from the default 10 minutes. Also, the SSS data expected to be received online by the plugin and sent to the CableDetector was not delivered. It was therefore not necessary with a new sea trial. It was later found that the REMUS interface is not compliant with the sonar. More results from this field test can be seen in Appendix C.2.

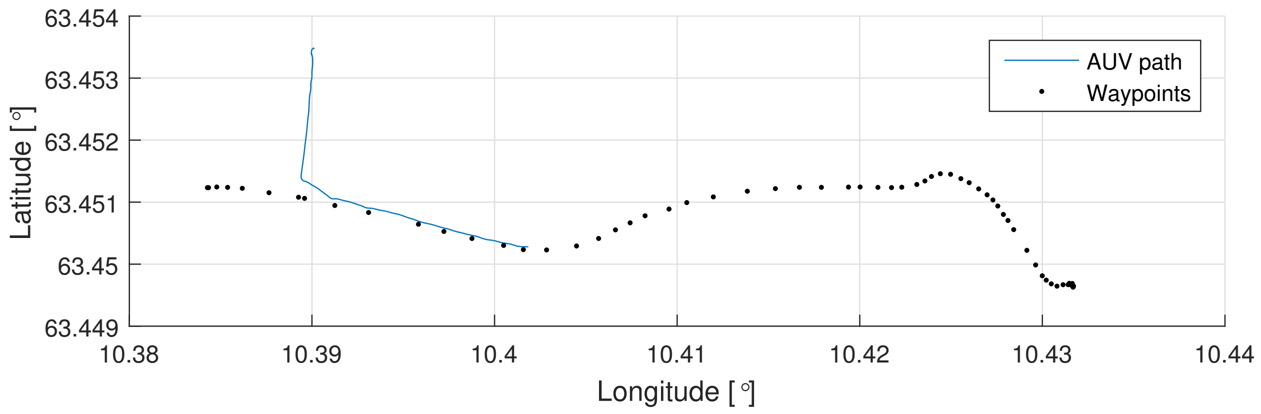


Figure 6.5: AUV position during sea trial March 29, 2017

6.2 Hemnfjord

In the late spring of 2017 we wanted to test the magnetometer on cables in Hemnfjord west of Trondheim in cooperation with Hemne Kraftlag SA. A pressure test of the magnetometer housing was performed June 1 which failed as it took in water, enough to sink the AUV. The sea trial was therefore postponed to June 15.

6.2.1 Preparations June 14

The Raspberry PI and magnetometers was successfully installed in the AUV housing shown in Figure 6.6 on June 14. Figure 6.7 shows the housing attached in the front end of the AUV. However, several challenges appeared:

- Too heavy magnetometer housing
- Might not have a cable to test on
- PP not communicating

The housing installed on the AUV was found to be too heavy making the buoyancy negative. Also, it was not confirmed that we could test on the cables at Hemnfjord as planned. These two problems can be fixed on a few days, but fixing the PP would require the AUV to be sent to a workshop and might take months. It was therefore decided to cancel the field test with the magnetometers since we were already past schedule.

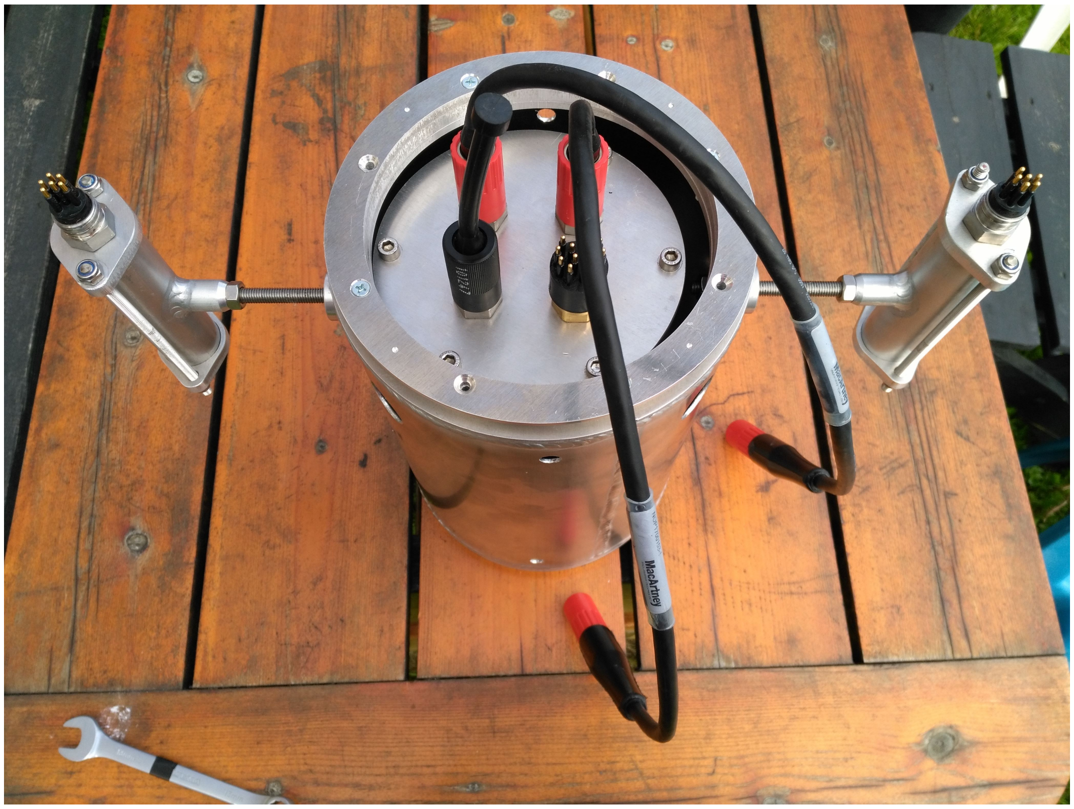


Figure 6.6: Housing with the magnetometers placed on the outside. A Raspberry PI is placed in the dry section in the middle

6.3 Simulations

Three types of data are used for sampling the states ϵ and ψ_{cab} during the simulations: a priori, SSS and magnetometer data. Note that the actual data itself is not sampled in the MR-EKF, but rather the estimates obtained by using the data.

6.3.1 A Priori Sampling

The a priori samples are obtained by using the AUV position and a priori data of the cable location. The position can be obtained through RECON when using the REMUS Simulator shown in Section 3.4 or by feeding the plugin with navigation data with time stamps obtained during sea trials. The states are then estimated as described in Section 4.1.4.

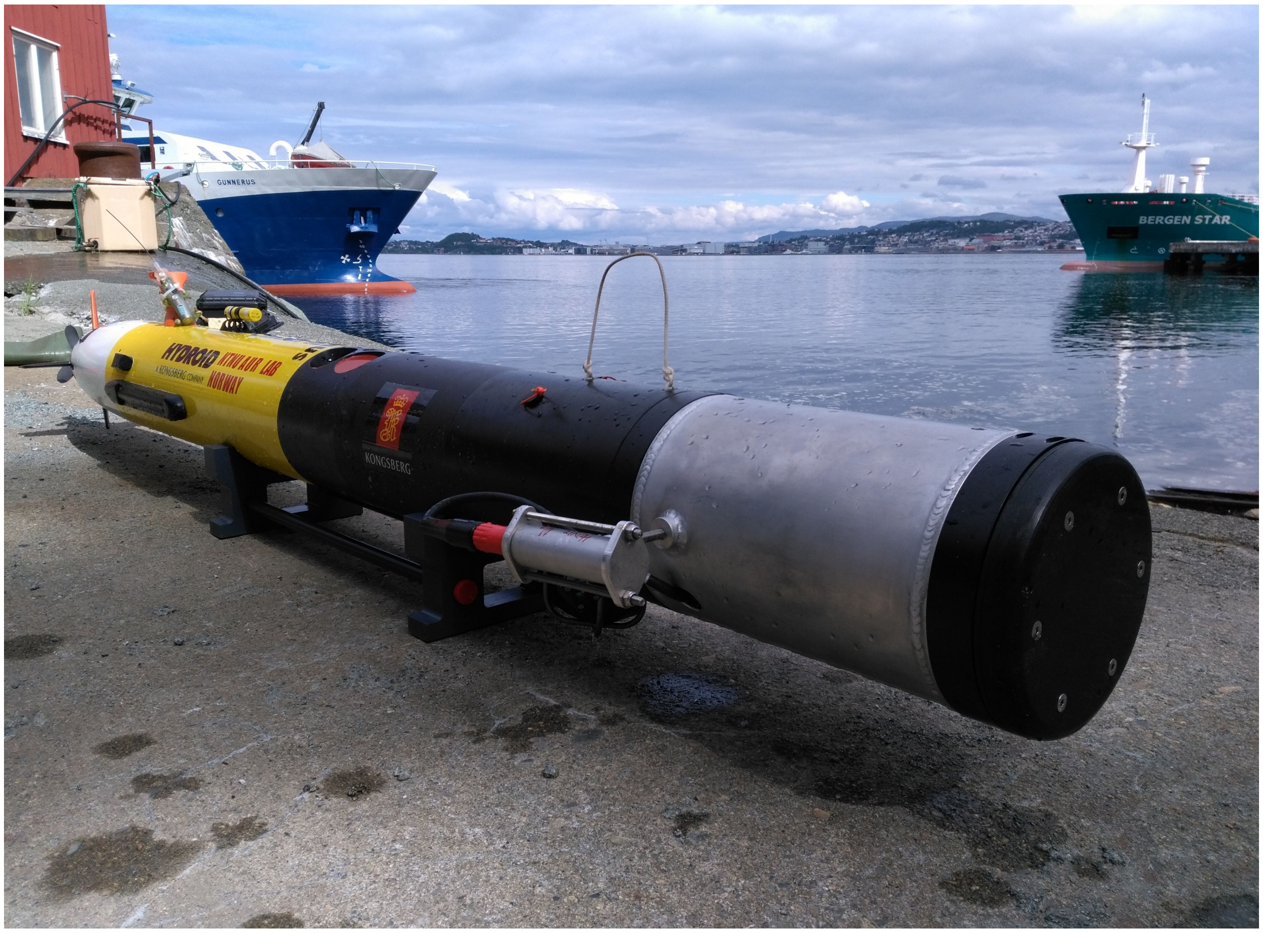


Figure 6.7: Remus with the installed magnetometer housing

6.3.2 Side Scan Sonar Sampling

The SSS samples are obtained by feeding the plugin with data retrieved after surveys. The data is synchronized with navigation data by using time stamps and sent to the Cable Detector which then provides estimations. The SSS samples are not used in combination with the REMUS Simulator because the position would not match.

6.3.3 Magnetometer Sampling

The magnetometer data is simulated since we have not had the opportunity to use the installed housing with the AUV. The TERE power cable at Munkholmen is simulated by using a priori data with noise and by not providing estimates when the AUV is far away from the cable. Per Kjetså (2017), this cable is a twisted 3 phase cable which we cannot easily detect because the three phases will reduce each other's magnetic field. We therefore assume it is a single AC cable with the return line far away. Kjetså has found that the cross-track error ϵ and relative heading ψ_{rel} can be provided for a cable with 100 A at an altitude of 4 m when ϵ is less than approximately 5 m. At distances up to about 30 m his code on the Raspberry PI will instead provide the plugin with information about which side of the AUV the cable is (starboard/port). At larger distances the Magnetometer data will not be used since the noise becomes larger than the magnetic field of the cable. Magnetometer data will only be used in the simulations when closer than 5 m to the cable, according to a priori data, since this would have given estimates of both states.

The Raspberry PI was able to give new estimations through UDP at a rate of 3 Hz during a test where the magnetometers were sampled at 25 Hz. The limitations are due to a large computational load relative to the limited processing power of the Raspberry PI. The states ϵ and ψ are therefore sampled in the plugin, from the magnetometer, at 3 Hz in the simulations.

Chapter 7

Results and Discussion

This chapter presents and discusses the results from the simulations and the sea trial of March 24, 2017. The data from March 29, 2017, will not be presented here since the mission was aborted, but some results can be seen in Appendix C.2.

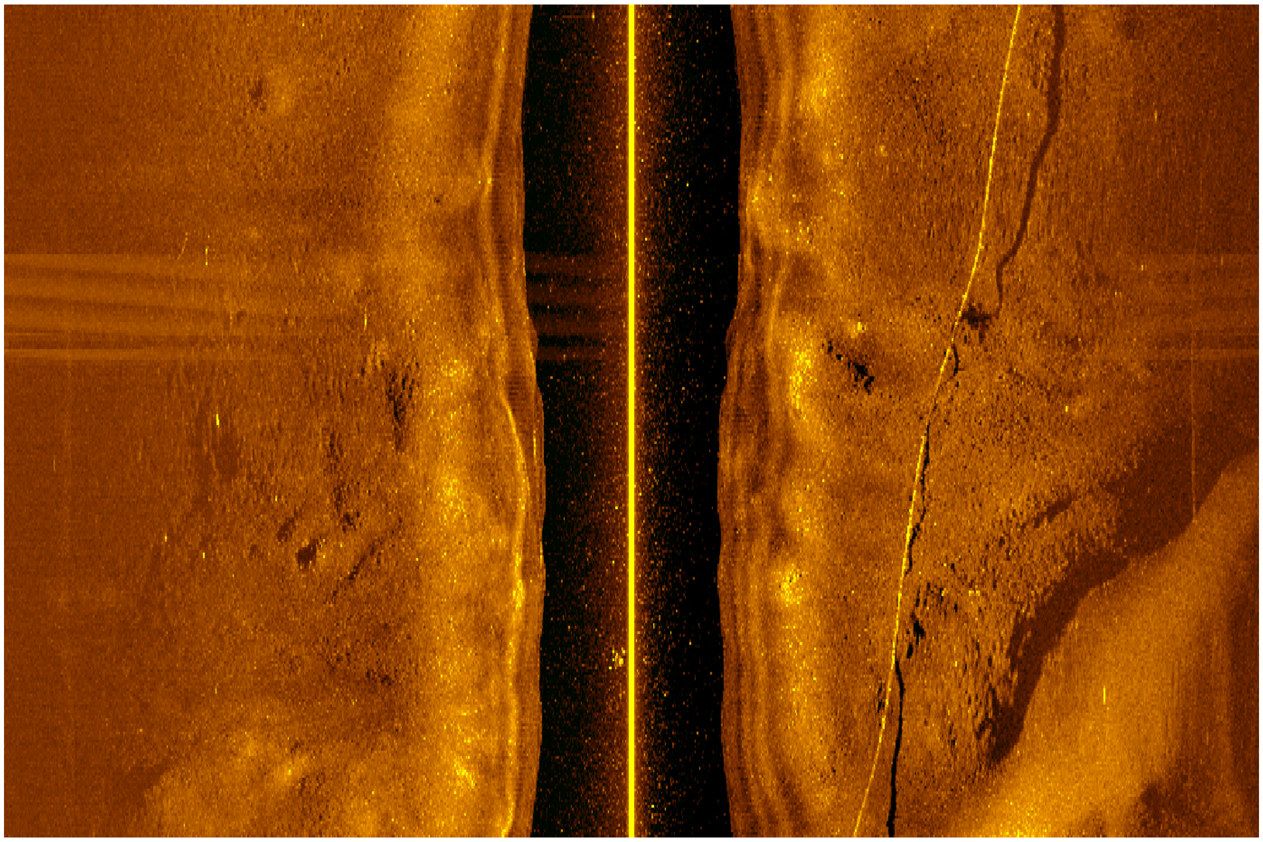


Figure 7.1: A SSS image exported from Sea Scan Survey showing a single cable on the starboard side of the AUV. March 24, 2017

7.1 Side-Scan Results

Figure 7.1 shows a SSS image processed and exported with the software Sea Scan Survey of Marine Sonics as a bitmap (BMP) image file. In this section the Cable Detector algorithm is tested by feeding it with one line (ping) at a time from the bottom of the 1388x921 image. The application was configured to use the previous 80 pings obtained from the plugin.

7.1.1 Offline Image Processing

Hough Transform

Figure 7.2 shows the accumulated weights from using SHT on a section of Figure 7.1. The darkest spots where the curves cross indicates distinct lines in the original image. Note that ρ moves clockwise from starboard as θ increases in the Cable Detector implementation. The highest value in the accumulator is marked with a circle having an θ of 18 ° and ρ of 270.5 pixels. This matches the cable quite exact and examples of detected lines can be seen on the next page.

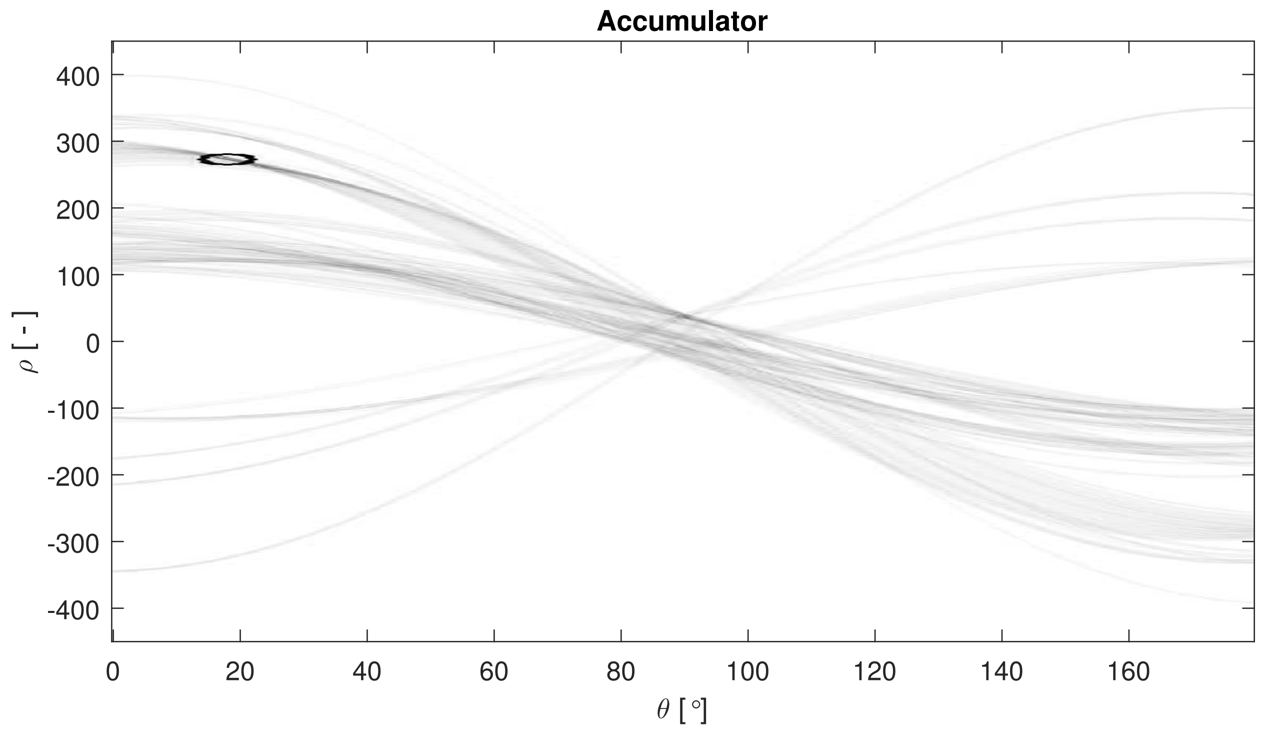


Figure 7.2: Hough weight accumulator with marked maximum

Cable Estimations

The frames in Figure 7.3 show the edge image and the detected line in the most recent pings from the BMP image. The cable is detected correctly in all of the frames presented except for frame 500 where no line is approved. Scaling is not applied in this test but an accumulator reduction of 0.65 is used and an angle resoluion of half a degree.

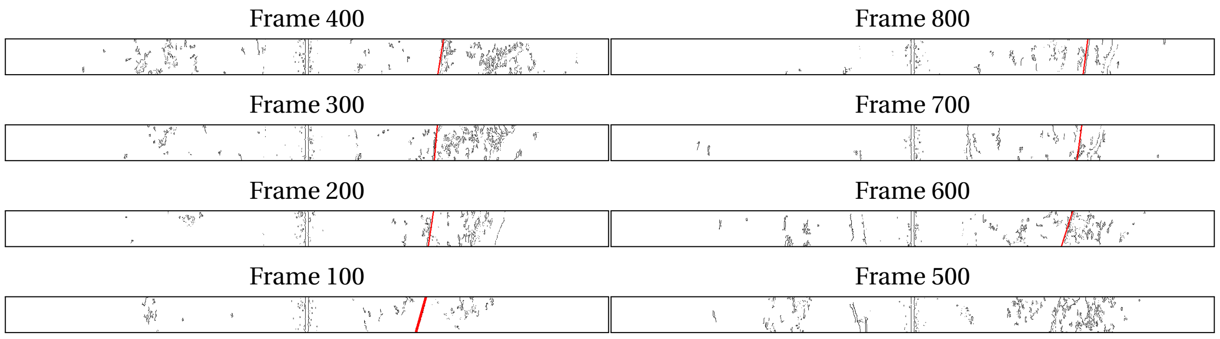


Figure 7.3: A selection of edge images with the detected line drawn in red

The line detection algorithm manages to find the cable and provide an accurate cross-track error ϵ and heading relative to the AUV ψ_{rel} shown in Figure 7.4 on the next page. The cross-track error is given in pixels since only information from the BMP image were used in this case. An accumulation threshold of 40 was set as a minimum value for a line to be approved but the cable estimations are fine before reaching the threshold in the beginning. The results show that unwanted measurements and spikes, but also a few good estimates, falls beneath this limit. The promising results can be used in LOS guidance if the cross-track error is converted to meters and the relative angle is corrected for scaling effects. By this we mean that the length per pixel in each direction will vary according swath width and speed. This is taken into account in the implementation for estimating the cross-track error when the cross-track error is converted into meters. However, the relative heading is just used as it is in this case. This could be solved by taking two points on the line and using their geographical coordinates to find the cable heading. We will instead scale the image before performing HT, from now on, although this is more computational demanding.

The last 80 pings are used, which takes about 10.6 s to obtain, because it gives good cable detection. This will however make the guidance algorithm vulnerable to changes in ψ_{AUV} as the cable heading is estimated as AUV pluss cable heading. The algorithm can be improved by

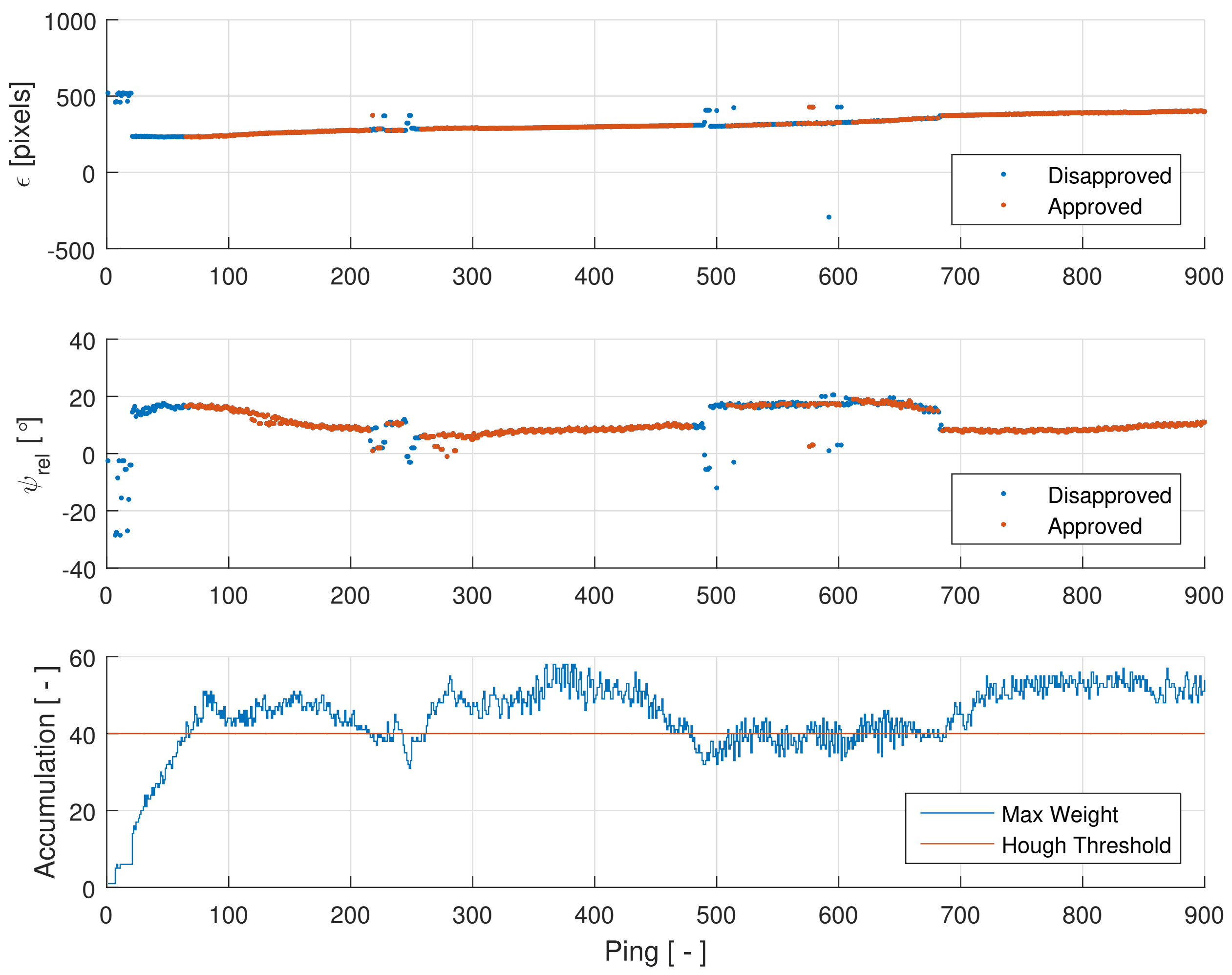


Figure 7.4: Cable estimations using the BMP image from Sea Scan Survey

using the ψ_{AUV} corresponding to the ping in the image where the cable is found. Another option is to decrease the number of pings used but this may give unsatisfactory results. The maximum angular rate set in the configuration will prevent sending results to the plugin during quick turns but not in the subsequent period when the angular rate is low and the image is distorted. One can remove the edge points from pings with high angular rates although this will lead to lower accumulated weights.

The algorithm showed good results on images with just water or flat grounds having none or few false positive results. However, some seabed areas with different structures were proven difficult. Removing edges from dark areas and shadows improved the results, but bright areas still make the Cable Detector falsely recognize cables. Thus, the program needs to be improved, used in combination with other input or only on a relatively flat seabed for online tracking.

7.1.2 Processing Time

The image processing time needs to be low to allow online cable detection and tracking. The algorithm speed was tested by processing 80 pings at a time from the image in Figure 7.1 on a Windows computer. By comparison with Figure C.1 in the appendix one can see that the processing time for the three methods is closely related to the number of edge points found in the image. This can also be seen in Figure 7.6 on the following page which shows a linear relation between the time used and the number of edges detected.

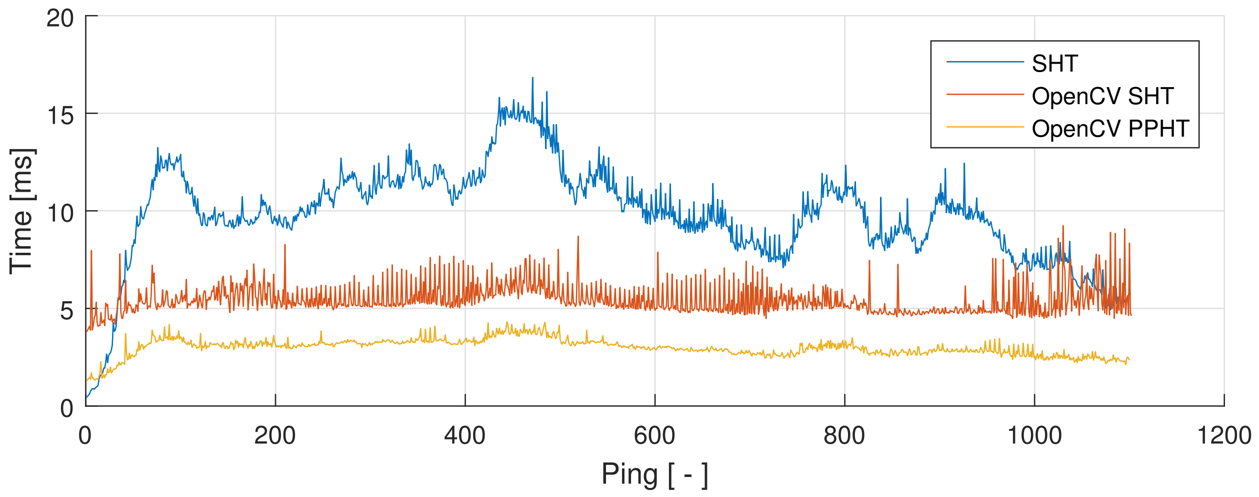


Figure 7.5: Hough Transform processing time

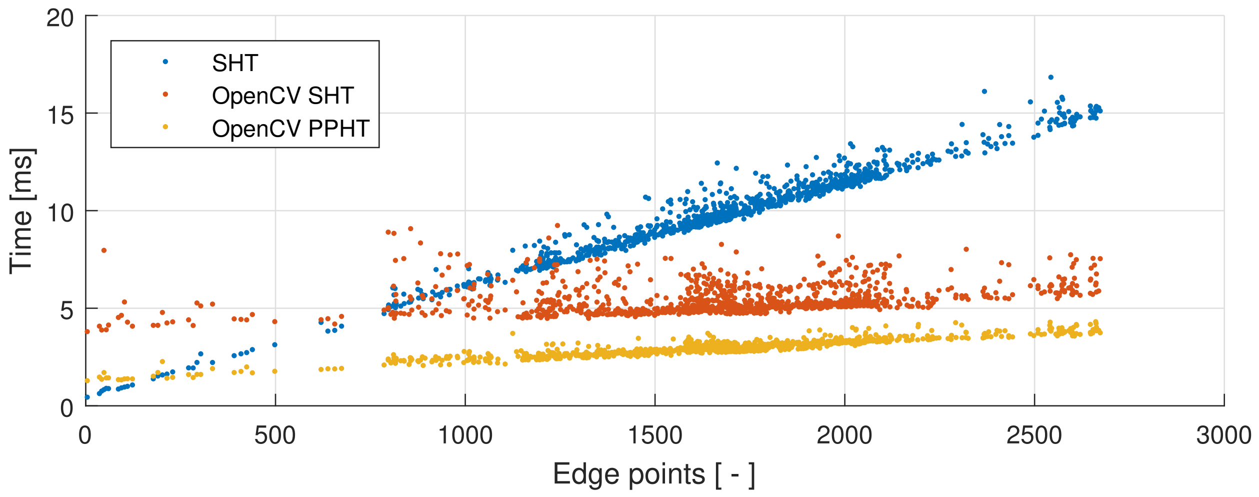


Figure 7.6: Hough Transform processing time as a function of edge points

Although the SHT algorithm has potential for improvement it is found to be good enough for the intended use on REMUS 100, even if the PP computer is slower. It is also possible to ease the computational demand by lowering the height of the image or lowering the angle resolution if the image processing is found to be slow. Another option to boost the performance is by using a separate computer with a dedicated graphics card. One can for instance use the specialized OpenCV HT functions for the CUDA platform by Nvidia.

The Cable Detector code needs to be fast if the solution is to be placed in the guidance feedback loop. This is because large time delays in ϵ make it very difficult to track the cable. Switching from the local state ϵ to two global states which do not depend on the AUV position would prevent the control from becoming unstable because of the processing time. This is why the absolute cable heading ψ_{cab} is used instead of the local relative heading ψ_{rel} in the state space.

7.2 The Effect of Applying TVG

Figure 7.7 shows a comparison of a SSS image before and after applying the TVG. The darker areas at longer range are brighter in the corrected image. Figure 7.8 shows the upper most ping in the image where the corrected ping has a more even intensity.

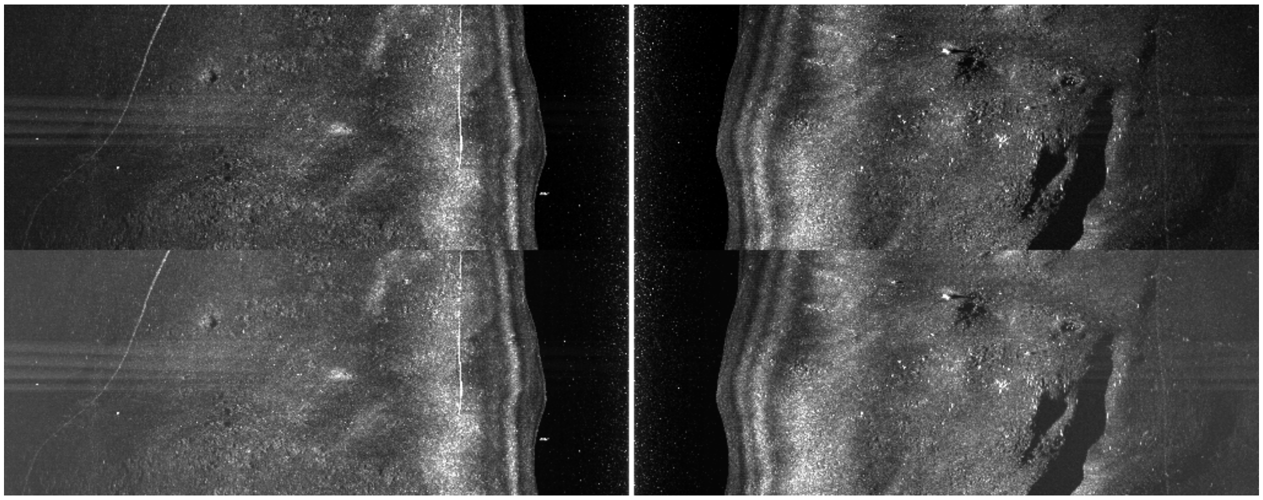


Figure 7.7: SSS image before and after applying TVG

A different line with more edge points is detected after applying TVG as seen in Figure 7.9. Neither of them is more correct than the other as both are parts of longer cable-like objects, but the Cable Detector should be able to find both.

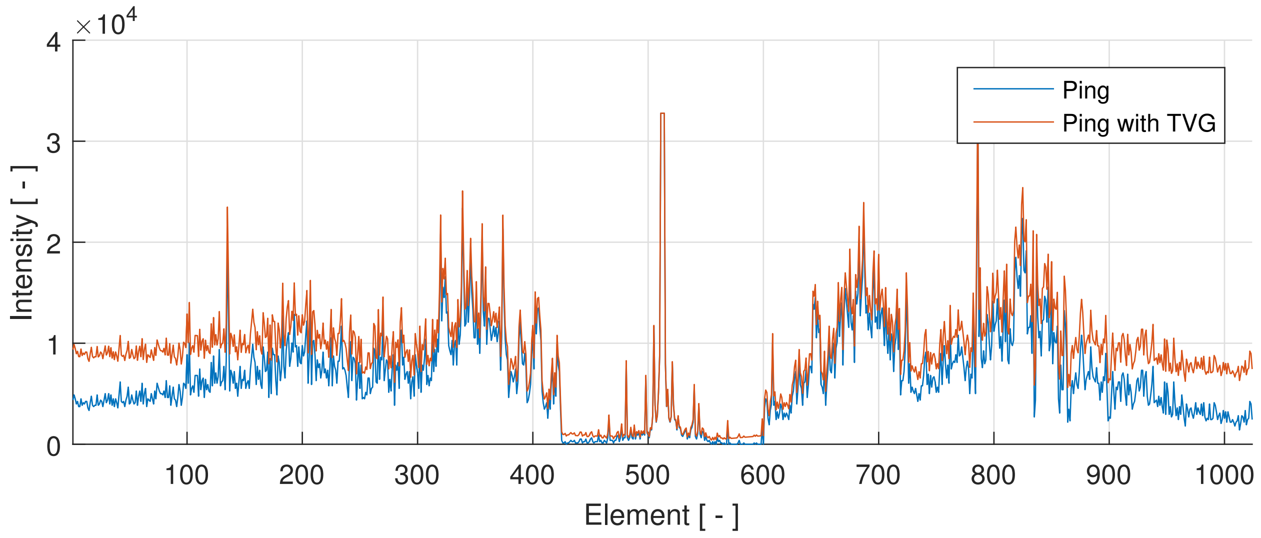


Figure 7.8: A single ping before and after applying TVG

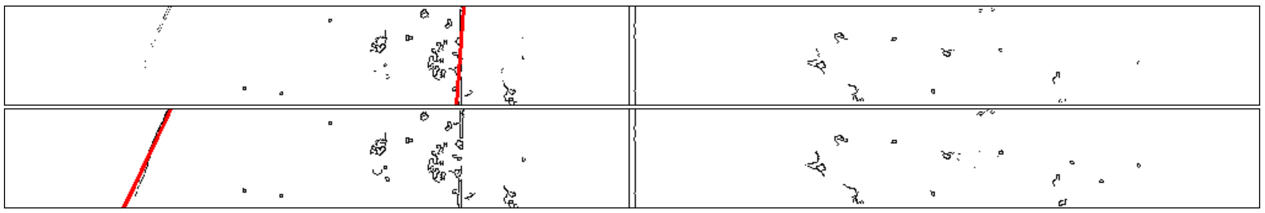


Figure 7.9: Most distinct line before and after applying TVG

The cables are easier to detect after applying the TVG, especially at larger ranges with higher transmission loss. This is because the cable is no longer disregarded for having low intensity. Cables lying relatively close to the AUV (7-15 m range) are still relatively difficult to detect as a result of the large intensity variations. Changing to a TVG based on realistic transmission loss estimates would improve the detection algorithm.

Some of the ping elements in the corrected image have a larger value than the maximum 32-bit value of 32767 and will be saturated. Although not many elements are affected in this ping, saturation will in general give an irreversible quality loss and should be avoided. This is left as possible further work along with changing the TVG.

7.3 Simulation Results using Multi Rate Extended Kalman Filter

This section shows the results of using the MR-EKF on a priori and SSS data, both independently and with sensor fusion. A simulation with imitated magnetometer sampling is also provided. The simulated measurement sampling is described in Section 6.3.

7.3.1 Estimations with A Priori Data

During the preparations for sea trials in Hemnfjorden the MR-EKF was tested with a priori data given by Hemne Kraftlag SA with a time step of approximately 40 ms. The filter covariance matrices Q and R in (7.1) and (7.2) were tuned until satisfactory results were obtained. $diag$ represents a diagonal matrix with the non-zero elements given in parenthesis. The first and the second values in the matrices represent the covariance for respectively the cross-track error in meters and the heading in radians. The a priori samples are cut between two waypoints to test the model, which lasted for about 28 s. The lookahead distance is set to 10 m as this gave good results in Østeby (2017) and the integral gain to zero for better stability.

$$\mathbf{Q} = \texttt{diag}(1, 1) \tag{7.1}$$

$$\mathbf{R} = \texttt{diag}(0.0001, 0.00001) \tag{7.2}$$

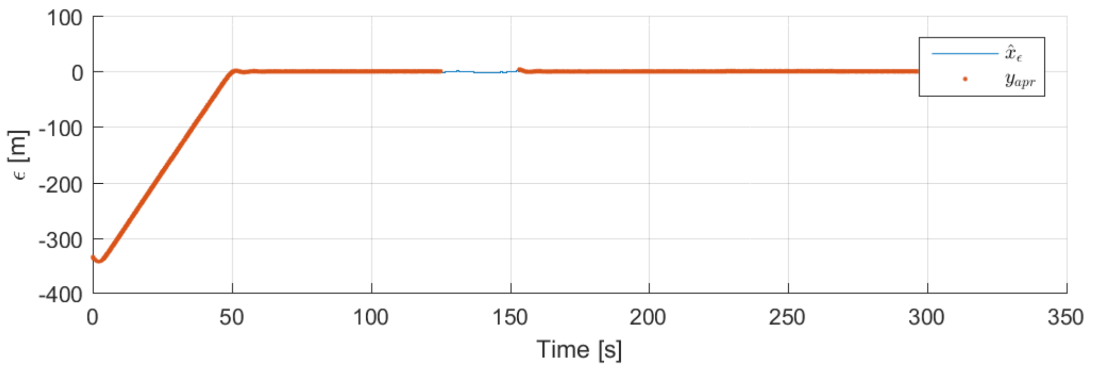


Figure 7.10: Cross-track error estimates using a priori data

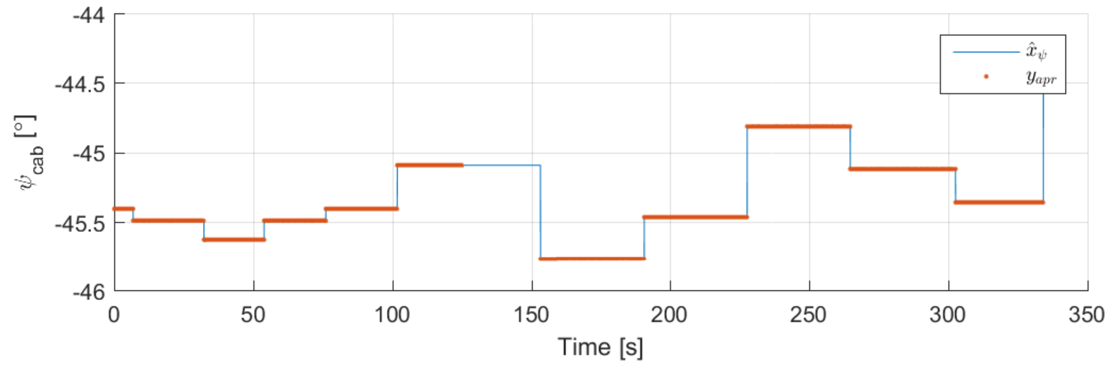


Figure 7.11: Heading estimates using a priori data

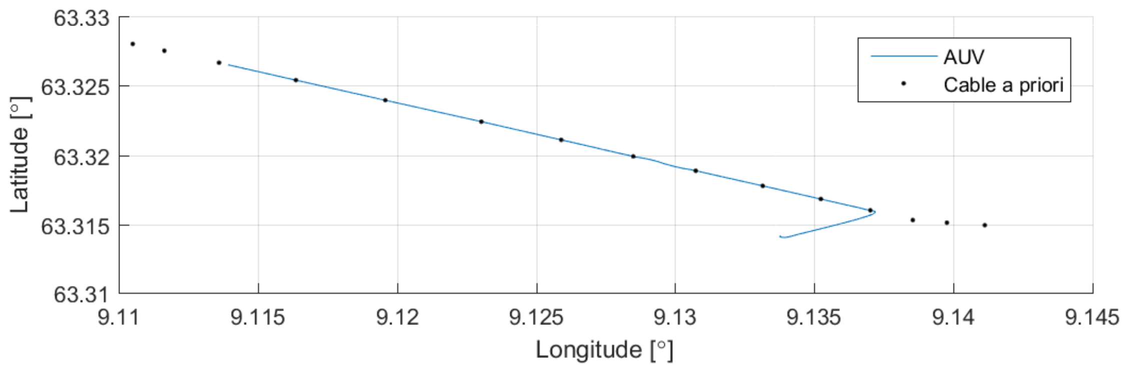


Figure 7.12: A priori cable coordinates and the AUV path in Hemnfjorden simulation

Figure 7.13 shows the estimated cable heading together with the AUV and LOS heading. The AUV follows the commanded LOS heading with a delay of up to 1 s to the cable with a small overshoot of about 1 m and manages to keep a low cross-track error. Losing the measurements led to oscillations of up to 3 m in the estimated $\hat{x}_\epsilon$ and 15° in the relative heading. The AUV heading becomes stable again when the measurements come back, but with a jump in ϵ of about 2 m.

Discussion

The AUV is capable of following the cable with and partly without measurements, although it is a bit unstable in the latter case. The instability issue might be fixed by tuning the lookahead distance and the covariance matrices. This is not prioritized because the small estimation error $\tilde{x}_\epsilon = x_\epsilon - \hat{x}_\epsilon$ and small ϵ are considered as good after a relatively long period without measurements.

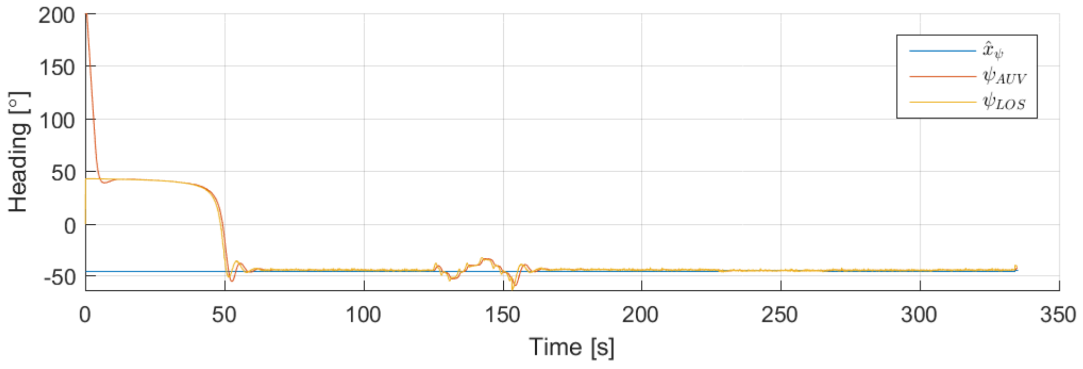


Figure 7.13: Estimated cable, AUV and LOS heading in Hemnfjorden simulation

7.3.2 Estimations with Side Scan Sonar Data

The results of only using SSS data to estimate the cable position are presented and discussed here, first from the Cable Detector and then from the MR-EKF. Munkholmen and Korsvika is almost at the same latitude and thus it is expected to find a cable heading close to 90° going from the island. The Cable Detector configuration can be found in B.2 on page 73.

Cable Detector Results

Figure 7.14 shows the results from processing the SSS images with an HT accumulation threshold of 40. The SSS files were stored by the sonar in four separate files because of the extent of the mission. We will focus on simulations with the third file which covers most of the interesting parts where the AUV is submerged. This means that the start time is 30 minutes into the mission. Figure 7.15 shows the estimated coordinates with the fourth file as well.

The Cable Tracker manages to find several line segments during the mission using the raw SSS data. The heading is mainly varying between 75° and 110° with some exceptions. The results also show that most of the scattered estimates are disapproved by having a weight below 40.

MR-EKF Results

The covariance matrices Q and R were again tuned and are shown in (7.3) and (7.4). The resulting estimations of feeding the filter with the Cable Tracker estimations are shown in Figure 7.16 and 7.17 for the cross-track error and cable heading respectively. The estimations are then used for calculating the LOS heading shown in Figure 7.18 using the same lookahead distance of 10

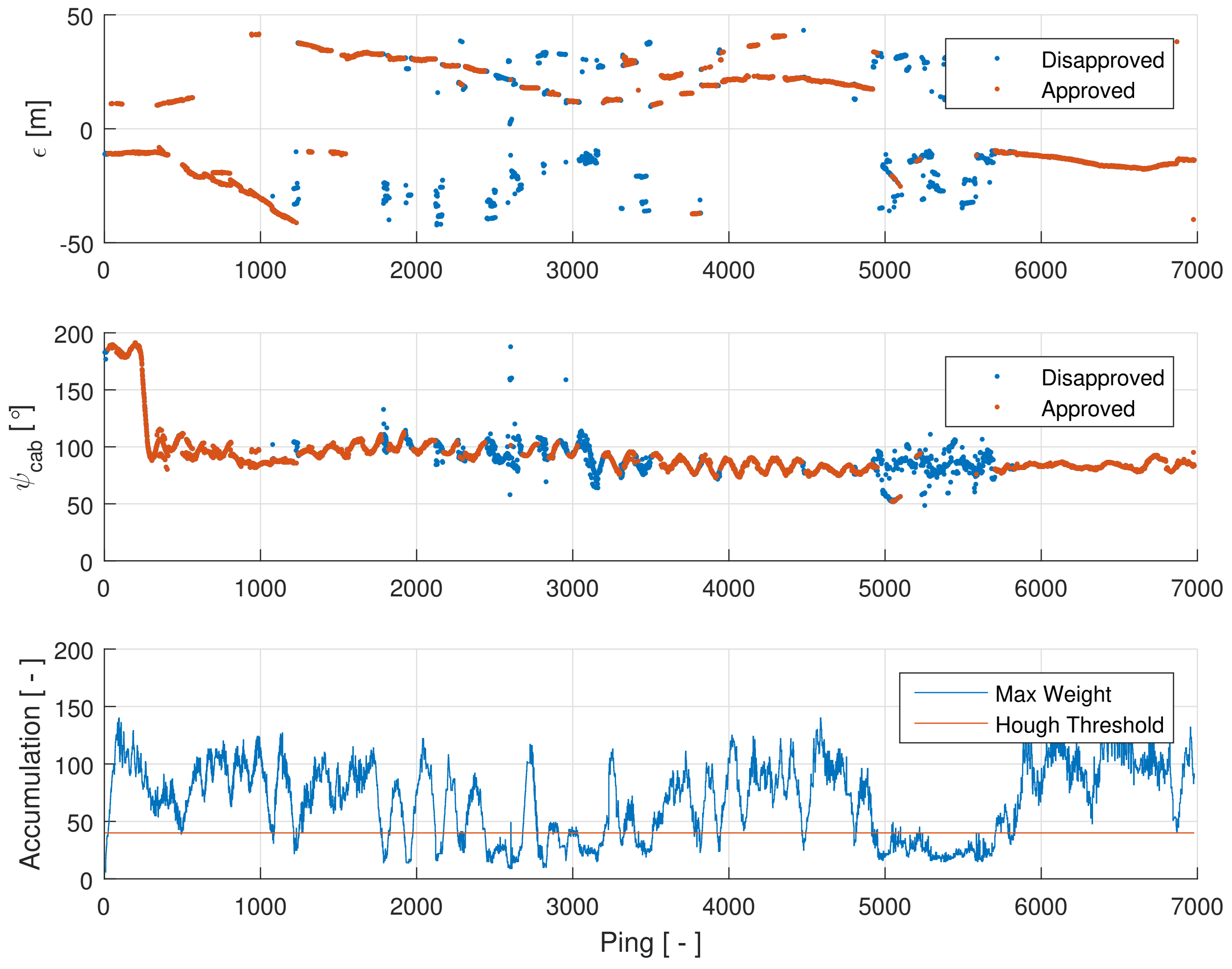


Figure 7.14: Cross-track error, cable heading and HT accumulation estimations from Cable Detector

meters and no integral gain.

$$\mathbf{Q} = \texttt{diag}(1,1) \tag{7.3}$$

$$\mathbf{R} = \texttt{diag}(0.1,0.001) \tag{7.4}$$

The resulting estimation $\hat{x}_\epsilon$ is stable at some parts but several jumps are observed. The missing measurements at ping 25-30 are removed by the program because of high angular rates. The measured cable heading is for the most part close to the AUV heading, and so is also the estimated $\hat{x}_\psi$. The calculated LOS heading is periodically stable, but has jumps in the same places as $\hat{x}_\epsilon$.

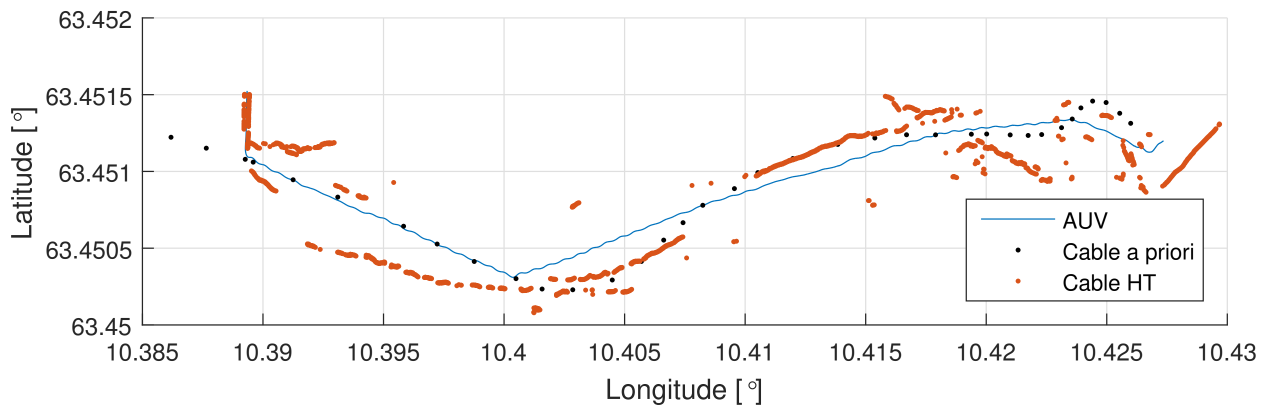


Figure 7.15: Cable position results from Cable Detector

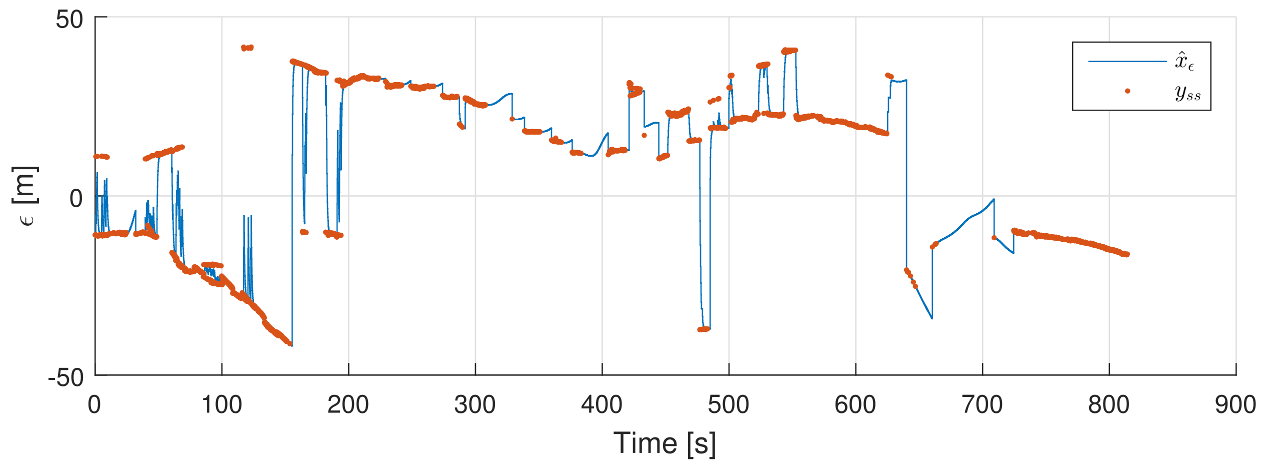


Figure 7.16: Cross-track error estimates using SSS data

Discussion

The estimated heading is close to the expected heading except in the first few hundred pings where reflections gave false positive estimates. Reflections from the surface cause symmetric lines in the image which are especially visible at lower depths and look similar to the cables. This makes the algorithm approve lines which are not objects in the beginning and at the end of the mission. To prevent false cable detection in the future, the program should estimate where the reflection line is by using the AUV depth as slant range and remove lines at this distance from the image. The solution used in the next simulations disregards all estimates when the reflection line is most prominent, which is up to approximately 8 m cross-track.

The oscillations in the cable heading is caused by oscillations in the AUV heading which can by comparing Figure 7.17 and 7.18. This shows the problem of only using the current heading

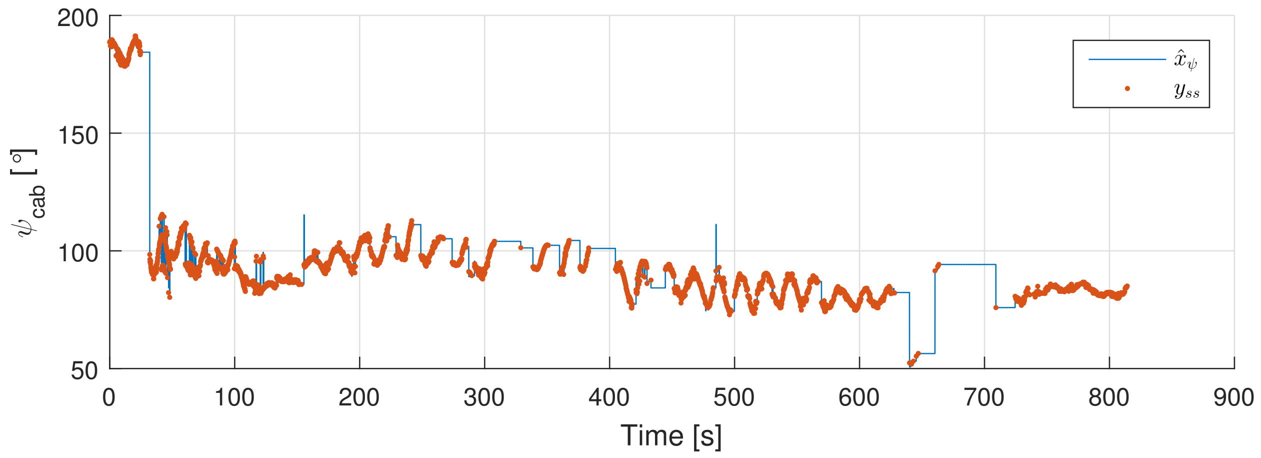


Figure 7.17: Cable heading estimates using SSS data

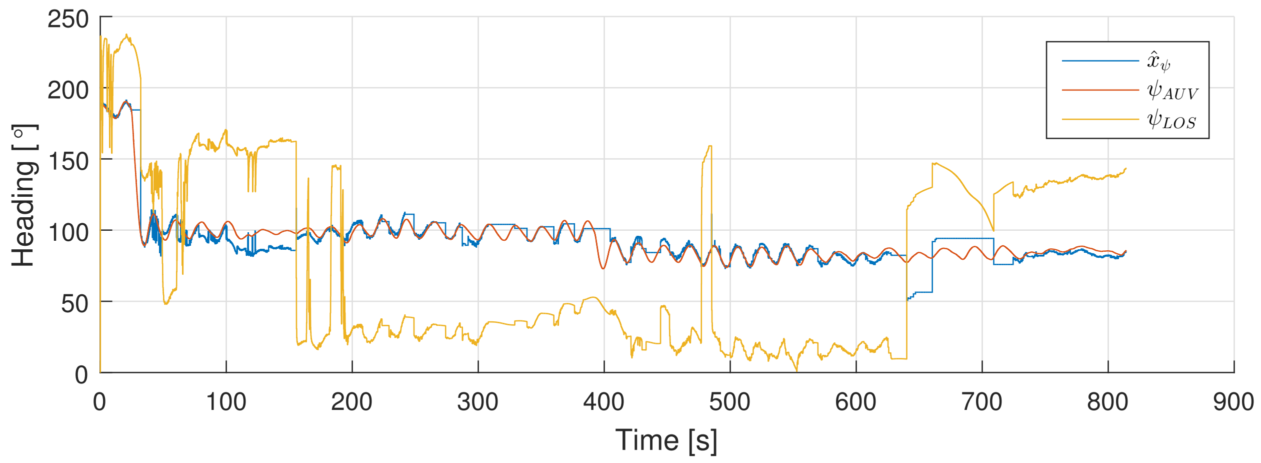


Figure 7.18: Estimated cable, AUV and LOS heading using SSS data

instead of the navigation information for each ping in the image. It also shows the need for new data where the AUV crosses the cable to see if the algorithm still works well.

The tracking algorithm is made for tracking a single cable and information will thus be lost when there is more than one cable. This is also a problem for the filter because multiple cable cause large variations in the measurements. To perform well in these cases the Cable tracker and/or filter needs to be changed. One can either implement a hysteresis for picking which cable to track or change the filter so that it can track multiple cables at a time.

The problem of having multiple cables can also be seen in the calculated LOS heading as this is a function of the cross-track error. Despite a few jumps in the LOS heading it does seem usable for guidance. The AUV does not follow this heading because it was calculated during simulation with survey data and not used in a feedback loop.

The cable position plotted in Figure 7.15 shows the estimated position of the cable by using the LOS parameters ϵ and ψ_{cab}. The coordinates may not actually lie exactly on the cable since the parameters stems from the HT polar coordinates. This can be the case if for instance the cable is turning or the closest point on the line lies outside the SSS image (in front). The PPHT by OpenCV provides line segments and can therefore be used for a more accurate plot of detected lines. However, lines are detected very close to the cable waypoints at several locations. This indicates that the estimates are not too bad and that the a priori data is accurate. The SSS is not meant for detecting things right beneath it and the AUV might be too close to the cable at the waypoints where no lines are detected. On the other hand, it did not help going about 5-6 meters further north as in Figure C.5 in the Appendix.

7.3.3 Combining Side Scan Sonar and A Priori Data

Here we look at the results of combining Cable Tracker estimations with a priori data in the MR-EKF with the same time step of 40 ms. The sample rate of the a priori data was set to 1 Hz while the SSS estimations were sampled if they arrived during the time step, meaning as fast as possible. The elements in the R matrix corresponds to ϵ and ψ_{cab} for the SSS and then for the a prior measurement covariance. The SSS data will be trusted more than the a priori data so that the AUV can track detected cables and the covariance for the latter in (7.6) is therefore set higher. The model is also trusted to get a more stable result. The Cable Detector results will be the same as in Figure 7.14 since the same SSS data is used.

$$\mathbf{Q} = \texttt{diag}(10^{-4}, 10^{-6}) \tag{7.5}$$

$$\mathbf{R} = \texttt{diag}(10^{-5}, 10^{-7}, 10^{3}, 10) \tag{7.6}$$

The results show that the estimated cross-track error $\hat{x}_c$ is much closer to the SSS samples y_{ss} than the a priori samples y_{apr} when a cable is detected. The estimate jumps when the Cable Detector finds a line at a different location and sometimes when no SSS samples are given. The cable heading estimate mostly follows the SSS samples. The LOS heading found in Figure 7.21

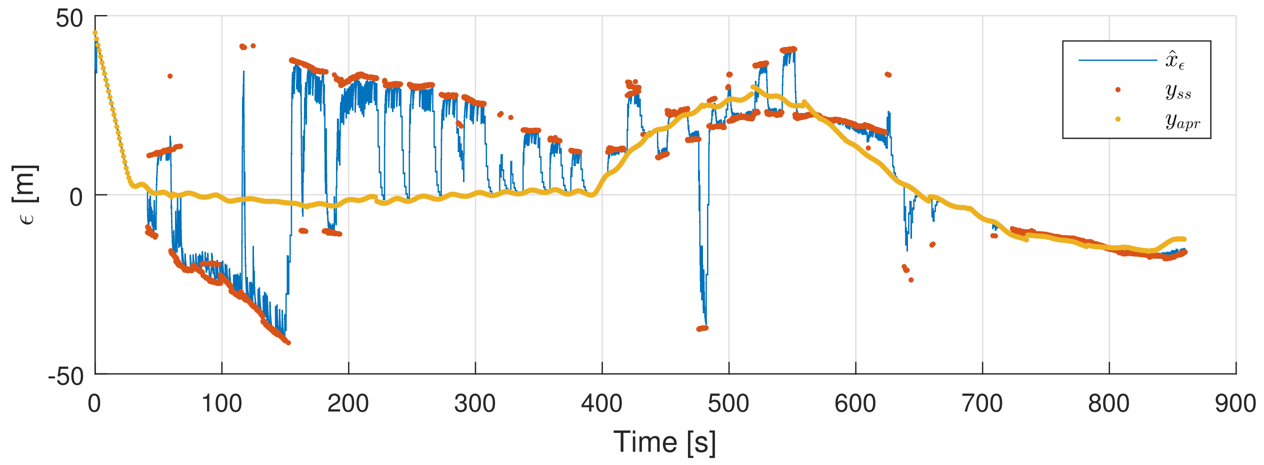


Figure 7.19: Cross-track error estimates using SSS and a priori data

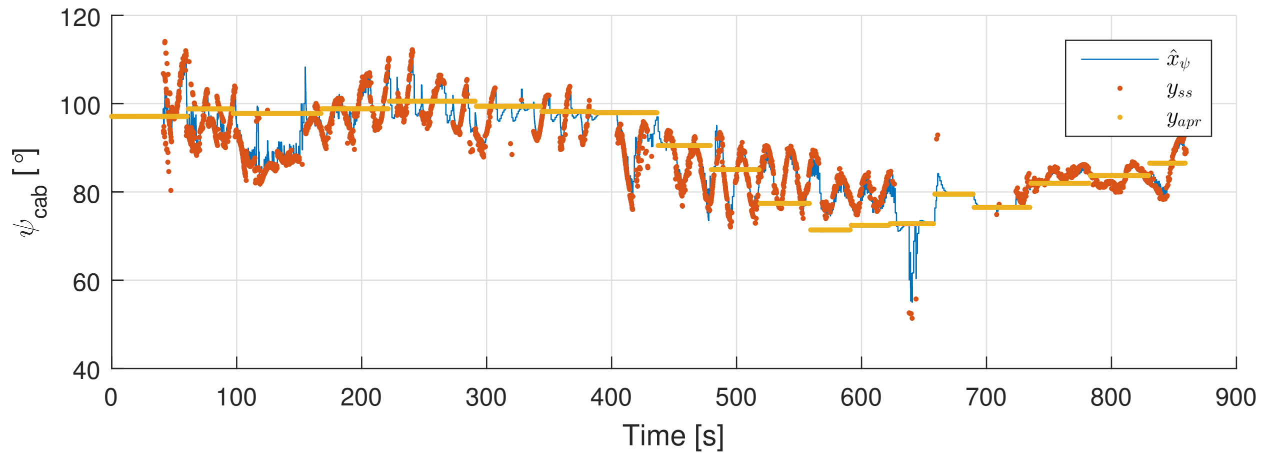


Figure 7.20: Cable heading estimates using SSS and a priori data

has some large variations when y_{ss} samples are missing.

Discussion

Since the same SS data is used we also get the same problem and possible solutions discussed in the last simulation about tracking multiple cables. However, now the a priori data affects the results and helps when no cable is detected in the period between 625 and 725 s. This comes at a price as the estimates change *from* the detected cable to the a priori estimate when SSS samples are missed. Since the quality of the a priori data will vary for different surveys the Q matrix should be tuned for each case.

The resulting LOS heading is promising but the large variations might cause trouble. A few

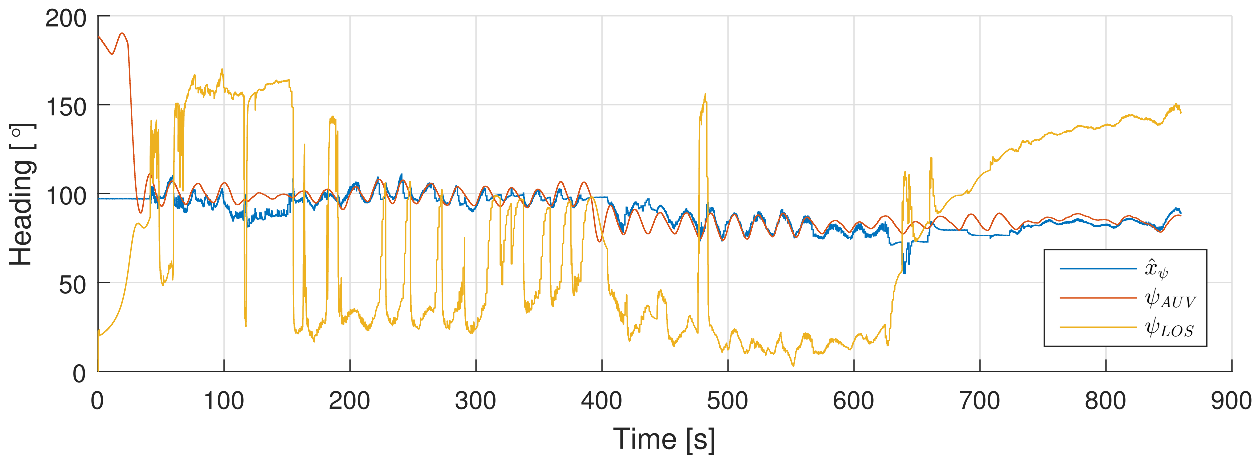


Figure 7.21: Estimated cable, AUV and LOS heading using SSS and a priori data

Spikes do not affect the guidance since the AUV do not have time to react. It is more problematic if the LOS has large variations and the AUV is constantly turning. Setting a higher lookahead distance will lower the LOS changes and thereby lowering the effect. Reducing the a priori sample rate or doing more tuning is also possible, but the main problem is that the cable is not always visible on the SSS image and seems to be partially buried. Having a magnetometer that could detect the cable would be of great help in this case.

7.3.4 Combining All Sensors

Here we combine SSS and a priori data with simulated magnetometer estimations. The covariance matrices in (7.7) and (7.8) were tuned until satisfactory results were obtained. The pairwise elements in the R matrix corresponds to the SSS, magnetometer and a prior measurement covariance respectively. This means that the magnetometers are trusted the most, then the SSS and the a priori data.

$$\mathbf{Q} = \mathtt{diag}(10^{-4}, 10^{-6}) \tag{7.7}$$

$$\mathbf{R} = \mathtt{diag}(10^{-4}, 10^{-6}, 10^{-5}, 10^{-7}, 10^{3}, 10) \tag{7.8}$$

Figure 7.22 shows that the estimated cross-track error at some periods is closer to the a priori measurements than in the previous simulation.

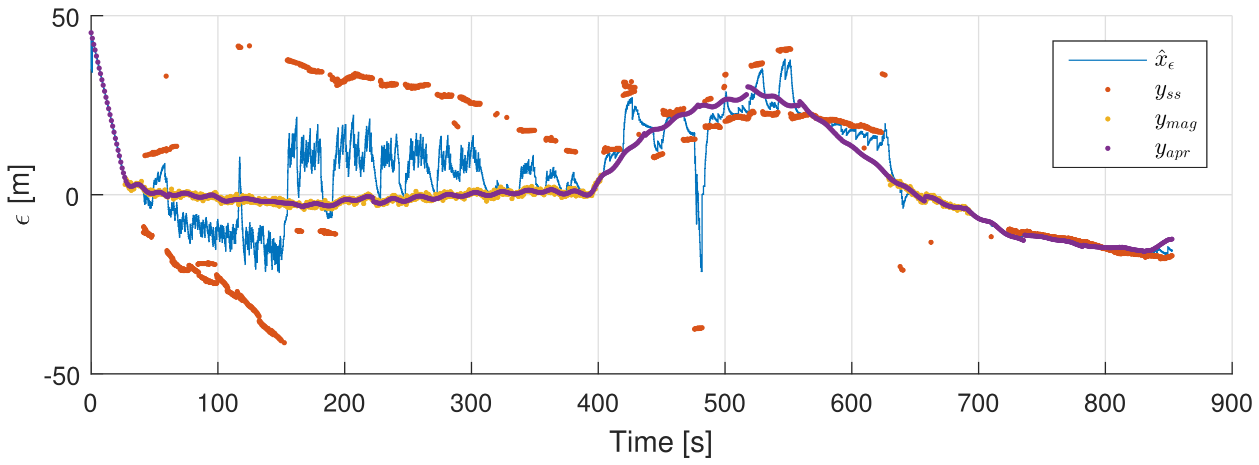


Figure 7.22: Cross-track error estimates combining all sensors

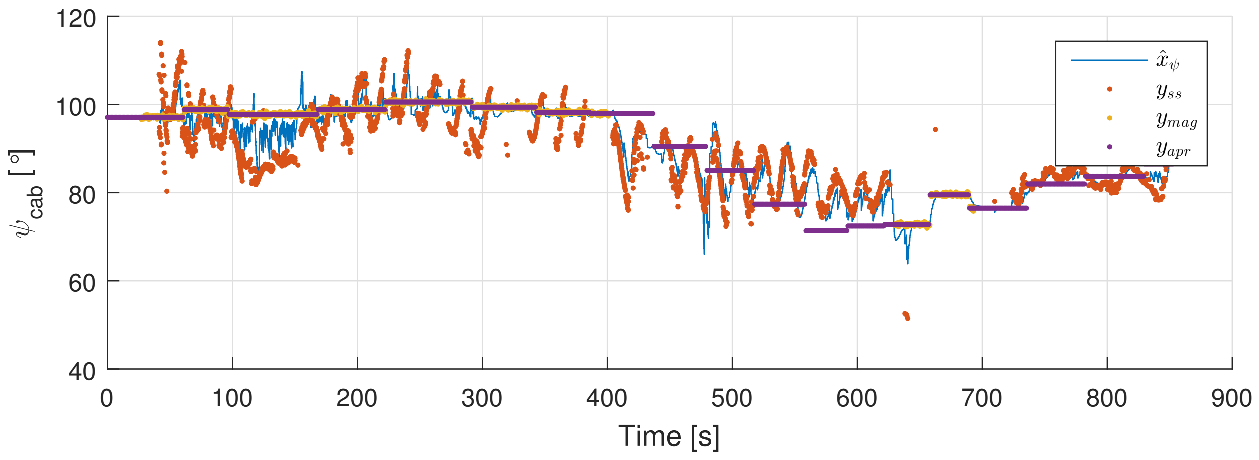


Figure 7.23: Cable heading estimates combining all sensors

7.3.5 Discussion

The simulated magnetometer measurements obtained when the AUV is closer than 5 m is found to affect the estimates. There is still problem of detecting several cables simultaneously and so the filter will find a middle way based on the covariance matrices. Again, this affects the LOS heading making it unstable when the cable is partially buried. Dropping the SSS samples when obtaining good magnetometer readings could potentially solve this issue so that the SSS will only aid the AUV at larger distances. Detecting and keeping track of several cables at a time is also possible as discussed earlier.

The very simple magnetometer sample model, which mainly is a priori data with noise, limits the learning outcome of this simulation. It shows that the MR-EKF is able to combine mul-

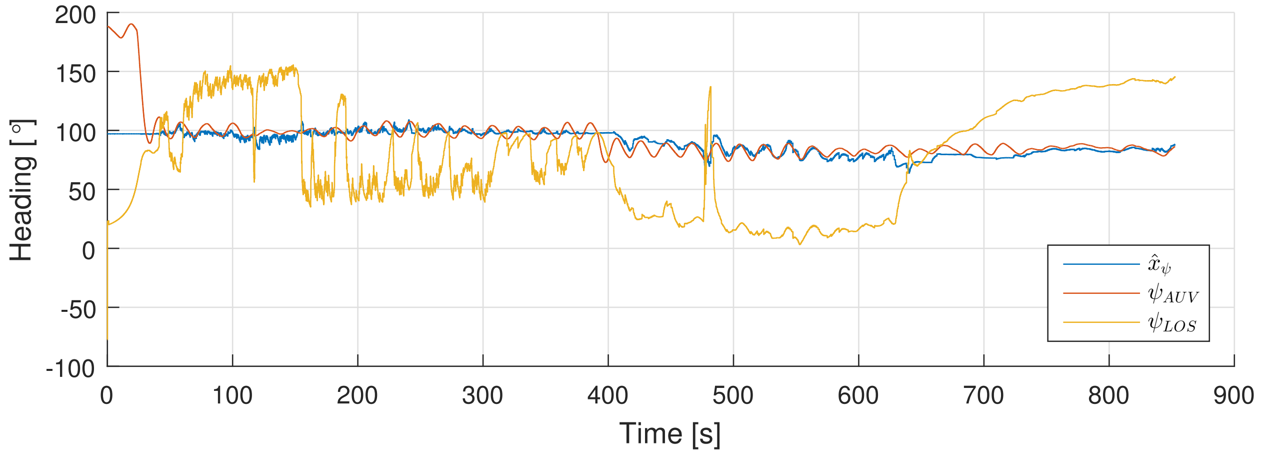


Figure 7.24: Estimated cable, AUV and LOS heading combining all sensors

tiple measurements with different sampling rates and can be tuned by changing the covariance matrices. However, it is advantageous if a more realistic model can be implemented or if the magnetometer housing is tested on a real cable before eventually improving the sensor fusion.

Chapter 8

Summary and Recommendations for Further Work

8.1 Summary and Conclusions

The results show how the developed program and plugin can be used to track relatively thin cables with a SSS when the cable visible or may be used in combination with other sensors and a priori data with a MR-EKF. The simple cable model used in the filter enables the AUV to estimate the cable cross-track error to the AUV in periods with missing measurements, assuming that the cable heading is the same. The estimates can then easily be used for LOS guidance in the plugin controlling the vehicle.

The Cable Detector program works by combining the raw SSS pings into an image. It scales the image before applying a TVG to compensate for transmission loss and a Gaussian filter to reduce noise. The Canny edge detector from OpenCV gave good results in finding edges in the image. Edges in dark areas are removed to prevent tracking shadows. The Cable Detector then uses HT to find the most prominent line or possibly cable. Using sensors for cable tracking in a feedback loop requires fast computation when using relative states such as the cross-track error. The HT is the most demanding image processing task in the algorithm and so the code was improved until it was considered fast enough for guidance.

The Cable Detector had difficulties differentiating between cables and seabed contours when looking at a small segment. A larger segment of 80 pings captured over approximately 10.6 seconds gave good results but makes the algorithm susceptible to changes in the AUV pose. It is currently assumed that the heading is constant in the last this period and set a high lookahead distance so that the heading changes relatively slow. The Cable Detector can be improved by utilizing the navigation information at each ping.

Tracking cables with images from Sea Scan Survey gave better results than with raw SSS data. This indicates that by processing the SSS data more, the tracking may be improved. This can be done by taking the AUV movement, especially pitching, into account and by improving the image filter.

The cable estimations were sometimes stuttering when combining the sensors in the simulation. One can implement other types of sensor fusion method to obtain smoother results as shown by Wang et al. (2014), but the main problem is that we expected one but found several cables. The current solution therefore works best on single cables.

Cable detection with magnetometers is already used by Statnett SF with ROVs since their submarine cables are mostly buried. However, the SSS solution can be very useful in areas where the cables create complex magnetic fields or when twisted three-phase cables cancel out the magnetic field. The solution is also applicable on many of the underwater cables in the grid where the operating voltage is 132 kV or lower. These are owned by smaller companies which only buries parts of the cable close to shore. Combining multiple sensors and prior information in the MR-EKF will therefore make the AUV capable of handling challenging tasks and perform the surveys more efficient than with ROVs.

8.2 Recommendations for Further Work

The first recommendation is to perform online cable tracking with magnetometers and SSS. This will reveal the potential of using magnetometers to detect underwater cables and if controlling the AUV by LOS will make it difficult to detect cables by SSS because of changes in heading. It would be a great contribution as there is a lack of scientific papers on online cable tracking by AUVs.

Moreover, it could be useful to test the Cable Detector with a SSS crossing a visible cable at different angles. The Cable Detector is currently tuned for detecting a single with the SSS data from Mars 24th 2017 and may not perform as well at other locations and scenarios without tuning or configuration. To handle tracking with multiple cables present it might be necessary with a selection method to decide which cable to track. Allowing the filter to estimate the position of multiple cables with global coordinates would also be advantageous to prevent large jumps in the estimates. Alternatively, constraints can be implemented by setting a maximum change in cross-track error over a short period of time or a maximum curvature which can be specified for each cable.

The Cable Detector can be improved by only removing surface reflections from the SSS image instead of discarding all of the lines. The estimated position of the reflection line can be used to remove points from the edge image, taking the altitude and depth at each ping into account. The results also indicated that there is potential for improvement in processing the image. Using the navigation information at each ping would enhance the performance by preventing the AUV movement from affecting the results. A more thorough comparison of different filters and gradient operators is also possible along with more parameter tweaking.

Appendix A

Acronyms

AC	Alternating Current
ADCP	Acoustic Doppler Current Profiler
AUV	Autonomous Underwater Vehicle
CP	Control Processor
DC	Direct Current
IP	Internet Protocol
LBL	Long Baseline
LOS	Line-Of-Sight
NavP	Navigation Processor
NTNU	Norwegian University of Science and Technology
PP	Payload Processor
ROV	Remotely Operated Vehicle
RPM	Rounds Per Minute
SDK	Software Development Kit
SOSI	"Samordnet Opplegg for Stedfestet Informasjon"
SSS	Side Scan Sonar

TDR	Time Domain Reflectometry
UUV	Unmanned Underwater Vehicle
UDP	User Datagram Protocol
RECON	REmote CONtrol
VIP	Vehicle Interface Program
VS	Visual Studio
WP	Waypoint

Appendix B

Configuration Files

The following shows configuration files used for the CableTracker plugin and the CableDetector program. A more thorough explanation of each parameter can be found in the actual files.

B.1 Cable Tracker Config File

```
# RECON Communication settings
[Communication]
UdpPort = 23456
ReconIP = 192.168.1.42

# Plugin Control settings
[Control]
StartLeg = 2
StopLeg = 10
DirectControlEnable = 0
DepthControlOnlyEnable = 0
SpeedCommandAsMetersPerSec = 1
EnableAltitudeControl = 1
AllowReconOnErrors = 1
DelayOverride = 20000
```

```
# Guidance algorithm parameters
[Guidance]
StartWaypoint = 7
StopWaypoint = 40
WaypointFile = CableMunkholmen.sos
CurveNumber = 1316084
Hemisphere = 1
ReturnToClosest = 1
DirectionCommandAsWaypoints = 1
LookaheadDistance = 10
IntegralGain = 0
SaturationLimit = 2000
OffsetNorthPerAltitude = 1
OffsetEastPerAltitude = 0
AltitudeCommand = 5
DepthCommand = 3
SpeedCommand = 1.5

# MR-EKF parameters
[SensorFusion]
UseMultiRateEKF = 1
FilterTimeStep = 40
NumberOfStates = 2
MatrixFolder = .\..\..\resources\KF_matrices_ss_apriori
FuseSideScan = 1
FuseMagnetometer = 1
FuseAprioriData = 1

# Cable Detector communication
```

```
[CableDetector]
NavPort = 8886
SideScanPort = 8887

# Raspberry PI communication
Port = 8884
```

B.2 Cable Detector Config File

```
# CableDetector settings
[CableTrackerPlugin]
NavPort =8886
SidescanPort =8887
CableTrackerIP =127.0.0.1

[ImageProcessing]
ArraysIncluded =80
PauseMillisec =5
SkipArrays =0
CableTres =170
UseCanny =1
CannyMinTres =100
CannyMaxTres =300
HoughTransform =0
HoughThreshold =40
HoughMinLineLength =50
HoughMaxLineGap =10
ThetaResolution =360
RhoResolution =1
AccumulatorReduction =0
```

```
SaveImages =0
TVGMultiplier =100
MaxAngRate =0.1
```

Appendix C

Additional Data

C.1 Offline Image Processing

C.1.1 Hough Transform Edge Points

Figure C.1 shows the number of edge points in the 80 last pings during simulation with data from March 24, 2017.

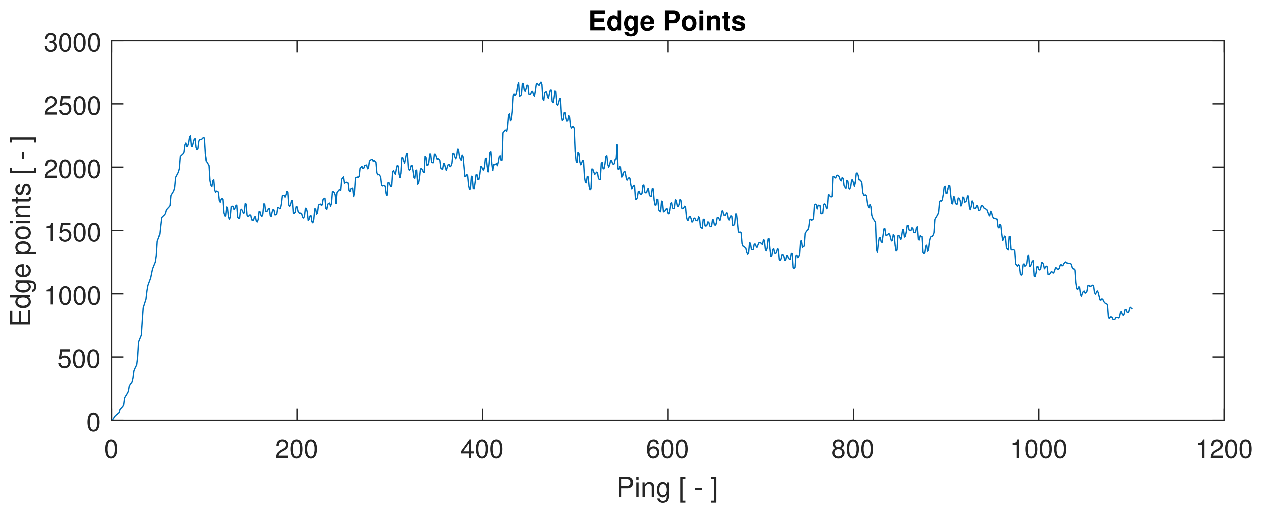


Figure C.1: Number of edge points from Canny function

C.1.2 Probabilistic Hough Transform

Figure C.2 shows an example of lines found with the OpenCV function called $HoughLinesP$, which uses an implementation based on the PPHT described in Section 2.2.4. The function has

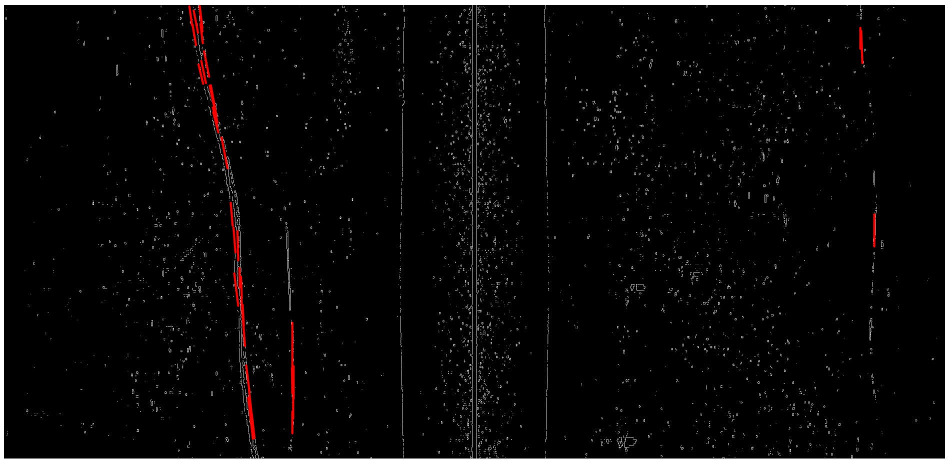


Figure C.2: Probabilistic Hough transformed image

the advantage of giving more exact coordinates for the detected lines instead of just a single polar coordinate. We can set the Hough threshold but we are not given the HT accumulation matrix and do not know which line got the highest weight.

C.1.3 Side Scan Sonar Mosaic Image

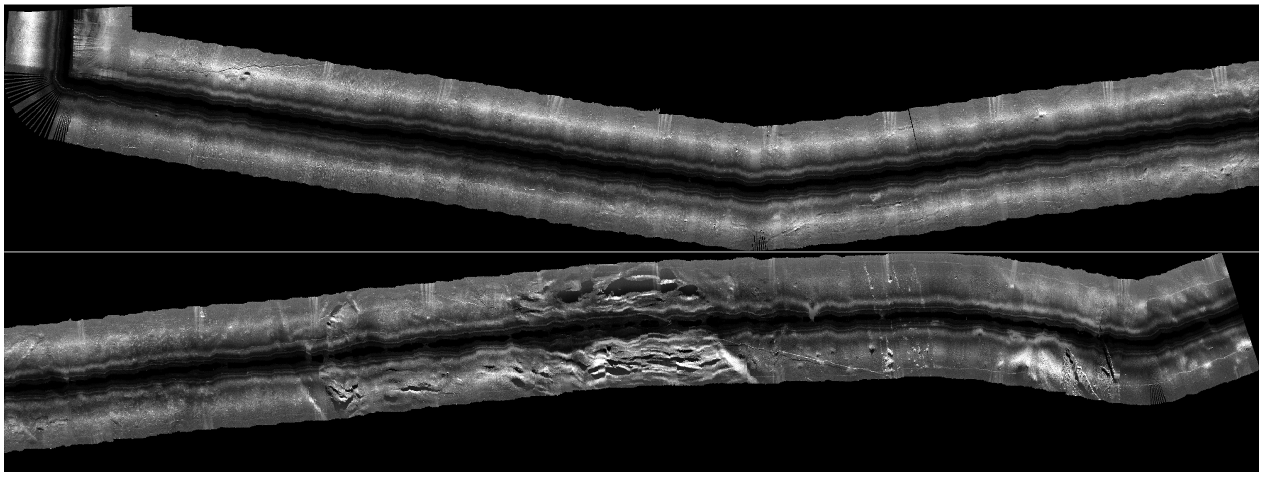


Figure C.3: Mosaic made with Reflection with data from March 24 2017

Figure C.3 shows a mosaic made with the software Reflection from the field trip March 24 2017. Some cable-like objects can be seen in the image.

C.2 Munkholmen Sea Trial March 29

Figure C.4 shows the results from the Cable Detector when simulation with data from March 29, 2017, when the mission was aborted after 10 minutes. AUV heading did not oscillate in this case when the plugin was used which provided straighter (more correct) cables in the SSS image. The plugin was configured to set an altitude of 5 m with an offset north of the cable equal to the altitude. The SSS range was set to 50 m and the speed to 1.5 m/s.

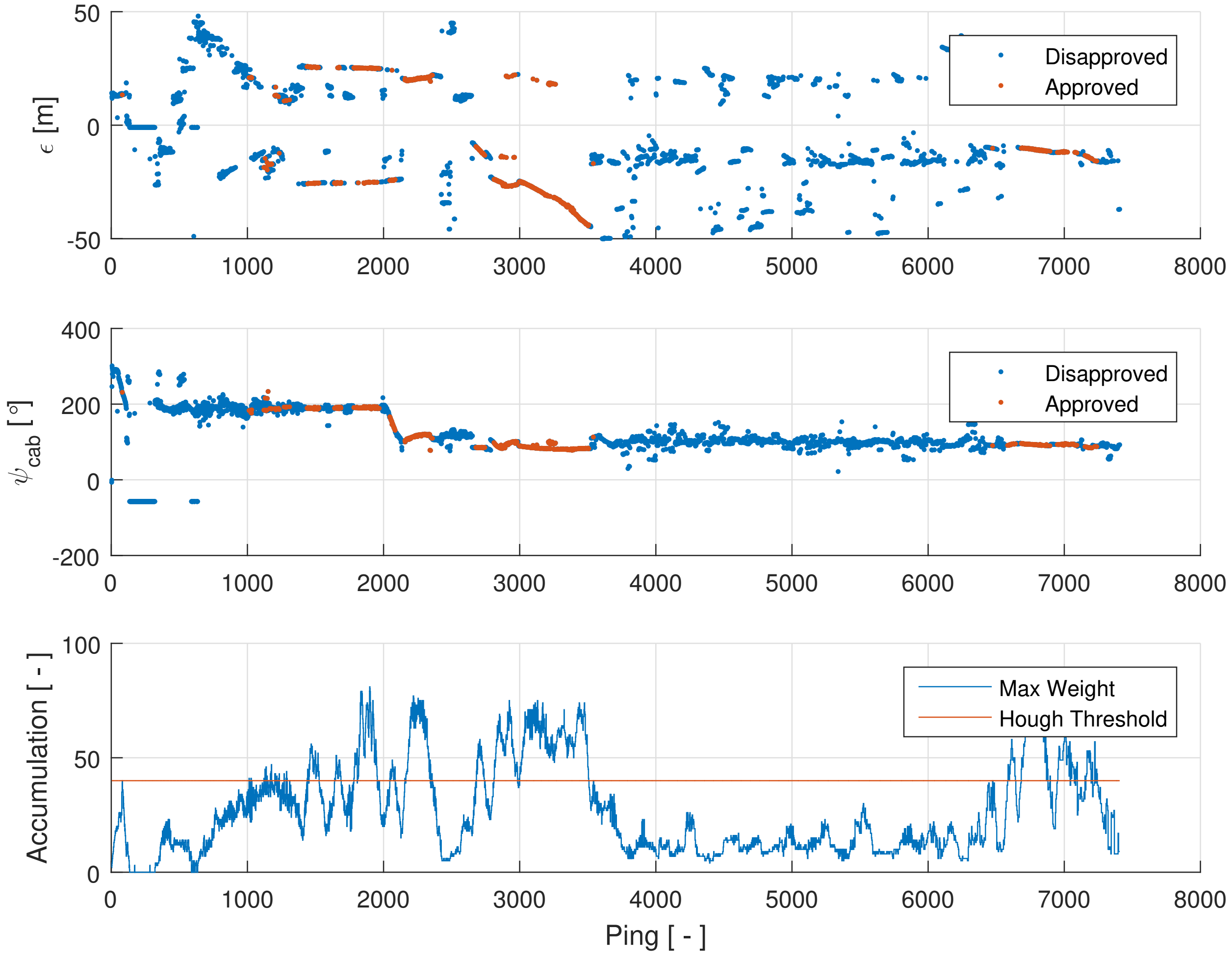


Figure C.4: Cross-track error, cable heading and HT accumulation from Cable Detector with data from 29. March

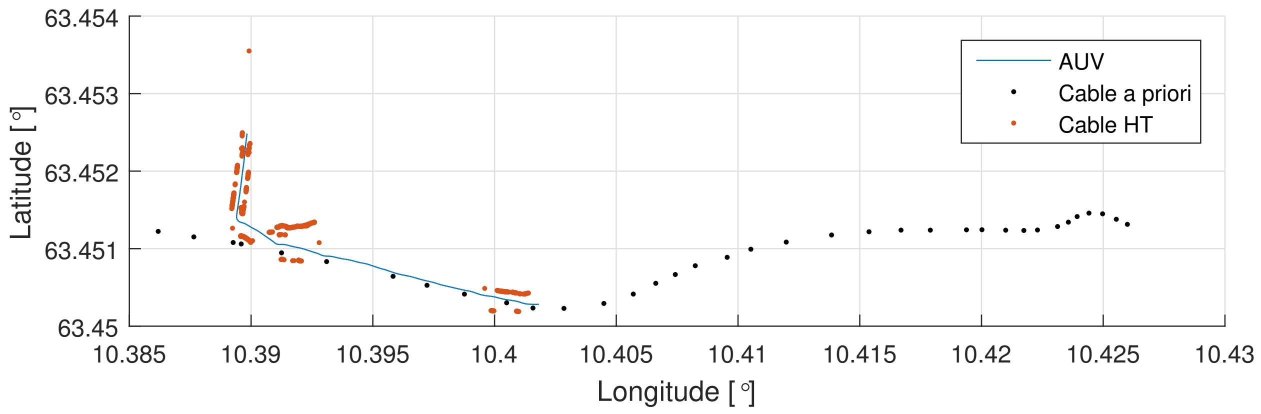


Figure C.5: Cable position results from Cable Detector with data from 29. March

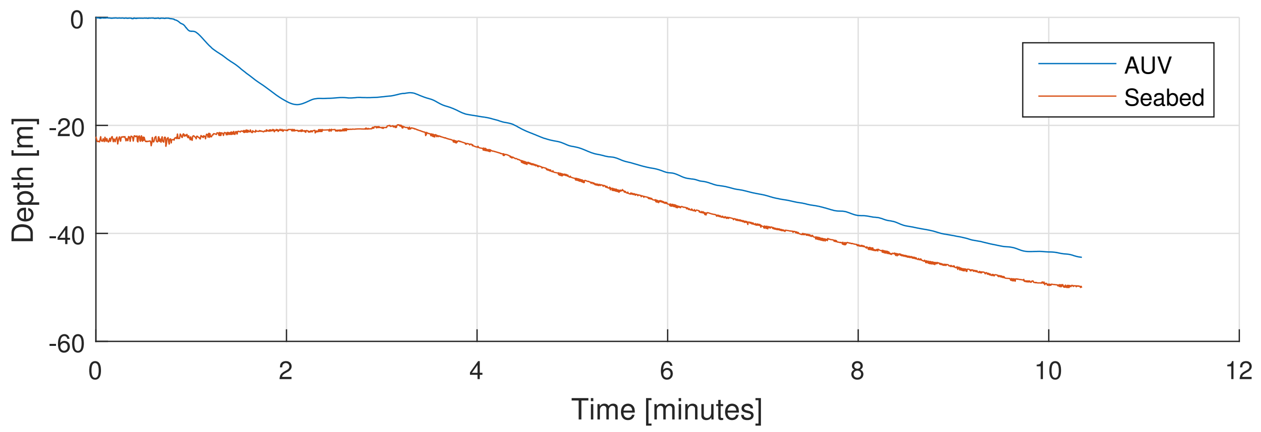


Figure C.6: AUV and seabottom depth until mission was aborted prematurely

The southern cable detected in the sea trial of March 24 was not as easy to find in this case as it was out of range for a longer period and at times not very visible.

Bibliography

Armesto, L., Tornero, J., and Vincze, M. (2007). Fast Ego-motion Estimation with Multi-rate Fusion of Inertial and Vision. *The International Journal of Robotics Research*, 26(6):577–589.

Asakawa, K., Kojima, J., Kato, Y., Matsumoto, S., Kato, N., Asai, T., and Iso, T. (2002). Design concept and experimental results of the autonomous underwater vehicle AQUA EXPLORER 2 for the inspection of underwater cables. *Advanced Robotics*, 16(1):27–42.

Bagnitsky, A., Inzartsev, A., Pavin, A., Melman, S., and Morozov, M. (2011). Side scan sonar using for underwater cables & pipelines tracking by means of AUV. In *2011 IEEE Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, pages 1–10. IEEE.

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.

Box, G. E. P. and Muller, M. E. (1958). A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics*, 29(2):610–611.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

Capus, C., Banks, A., Coiras, E., Tena Ruiz, I., Smith, C., and Petillot, Y. (2008). Data correction for visualisation and classification of sidescan SONAR imagery. *IET Radar, Sonar & Navigation*, 2(3):155.

Draka Norsk Kabel AS (2010). TERE 1kV Datablad.

Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*, volume 17. John Wiley & Sons, Ltd, Chichester, UK.

Hagen, P. E. and Kristensen, J. (2002). The HUGIN AUV "Plug and play" payload system. In *Oceans '02 MTS/IEEE*, volume 1, pages 156–161. IEEE.

Hovem, J. M. (2012). Marine Acoustics : The Physics of Sound in Underwater Environments.

Hydroid (2010). REMUS 100: Operations & Maintenance Manual.

Innovatum Ltd (2012). INNOVATUM PRODUCT REFERENCE Section 0. (2):1–14.

Kälviäinen, H., Hirvonen, P., Xu, L., and Oja, E. (1995). Probabilistic and non-probabilistic Hough transforms: overview and comparisons. *Image and Vision Computing*, 13(4):239–252.

Kjetså, A. (2017). Submarine Power Cable Localization by REMUS 100 AUV.

Ludvigsen, M. (2016). Lecture Notes in Arctic Marine Measurements Techniques, Operations and Transport.

Matas, J., Galambos, C., and Kittler, J. (1998). Progressive Probabilistic Hough Transform. In *Procedings of the British Machine Vision Conference 1998*, volume 24, pages 26.1–26.10. British Machine Vision Association.

Menconi, P. F. and Liparoto, D. J. (1992). TRACKING AND FAULT LOCATION IN UNDERSEA CABLES. In *OCEANS 92 Proceedings. Mastering the Oceans Through Technology*, volume 2, pages 678–683. IEEE.

Østeby, E. (2017). *Cable Tracking by the Autonomous Underwater Vehicle REMUS 100.* Specialization project, NTNU, Trondheim.

Petillot, Y., Reed, S., and Bell, J. (2002). Real time AUV pipeline detection and tracking using side scan sonar and multi-beam echo-sounder. In *Oceans '02 MTS/IEEE*, volume 1, pages 217–222. IEEE.

Postel, J. (1980). User Datagram Protocol.

Szyrowski, T., Sharma, S. K., Sutton, R., and Kennedy, G. A. (2013a). Developments in subsea power and telecommunication cables detection: Part 1 - Visual and hydroacoustic tracking. *Underwater Technology*, 31(3):123–132.

Szyrowski, T., Sharma, S. K., Sutton, R., and Kennedy, G. A. (2013b). Developments in subsea power and telecommunication cables detection: Part 2 - Electromagnetic detection. *Underwater Technology*, 31(3):133–143.

Wang, Y., Kostic, D., Jansen, S. T. H., and Nijmeijer, H. (2014). Filling the gap between low frequency measurements with their estimates. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 175–180. IEEE.

Xiang, X., Yu, C., Niu, Z., and Zhang, Q. (2016). Subsea cable tracking by autonomous underwater vehicle with magnetic sensing guidance. *Sensors (Switzerland)*, 16(8):1–22.

Xu, L., Oja, E., and Kultanen, P. (1990). A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338.