

# Line-of-sight iceberg edge-following using an AUV equipped with multibeam sonar

Petter Norgren and Roger Skjetne

*Department of Marine Technology, Norwegian University of Science and Technology (NTNU), {petter.norgren, roger.skjetne}@ntnu.no.*

**Abstract:** Obtaining 3D information about ice features, like icebergs, are of interest to researchers and offshore operators moving into the Arctic. Icebergs are affected by wind, and ocean currents, and can have unpredictable drift patterns, causing challenges when it comes to mapping objectives. Autonomous underwater vehicles (AUVs) equipped with multibeam echosounders are suitable for obtaining measurements of the underwater geometry of icebergs, but advances in autonomy are needed to map drifting icebergs reliably. This paper details a guidance algorithm for detecting and circumnavigating an iceberg – following the iceberg edge. The guidance scheme is implemented as a state machine, starting in an *iceberg detection*-mode. Once an iceberg is detected, the AUV will enter a *mapping*-mode. An edge detection algorithm will determine the position of the edge, and a line-of-sight approach will be used for edge-following. A six degree-of-freedom AUV simulator is used to perform a simulation study, to show how AUV dynamics affect the results. The simulation study presented shows the algorithm's effectiveness, both when the iceberg is assumed stationary, and when the iceberg is drifting and rotating with constant velocity.

© 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

*Keywords:* Autonomous underwater vehicles; Line-of-Sight guidance, Iceberg mapping.

## 1. INTRODUCTION

In Arctic offshore activities, where there may exist a threat of sea-ice, and/or icebergs, ice management (IM) must be employed. Eik (2008) defines IM as the sum of all activities where the objective is to reduce, or avoid, actions from any kind of ice features. This includes detection, tracking, and forecasting of sea-ice, ridges, and icebergs. Obtaining information about the current local ice conditions is crucial for the decision-making process in an IM operation. Lack of (or inaccurate) information may lead to hazards, or downtime due to unnecessary disconnects.

Autonomous underwater vehicles (AUVs) are a class of underwater vehicles, characterized by having relatively high spatial and temporal coverage, that are unaffected by the potentially harsh surface conditions in the Arctic. Norgren and Skjetne (2014) discusses the use of AUVs as a sensor platforms for ice monitoring applications, and concludes that AUVs are suitable for ice-monitoring, under the requirements of available infrastructure, and further research within the field of autonomy.

From the first reported AUV deployment in the Arctic in 1972, presented by Francois and Nodland (1972), more and more complex AUV missions have been conducted in the Arctic. Wadhams et al. (2004) presents the use of a Maridan Martin 150 AUV that gathered side-scan imagery of the underside of the ice in 2002 – the first of its kind acquired by an AUV. From the side-scan sonar data, the authors were able to identify first-year, multi-year, brash, and frazil ice. Similarly, the Autosub-II AUV was used to obtain the first under-ice multibeam measurements in 2004 (Wadhams et al., 2006). These missions demonstrate

the AUV's unique capability of being able to survey the underside of the ice over large areas.

A previous experiment aiming to map icebergs by means of an AUV was presented by Forrest et al. (2012), where four AUV missions was conducted under a fragment of the Petermann Ice Island at a depth of 60 m. During these missions, draft measurements from the underside of the iceberg was captured using an interferometric sidescan sonar. Multibeam measurements of the side of the iceberg keel was also captured, from a ship-mounted multibeam echosounder (MBE). Forrest et al. (2012) states that one of the biggest challenges encountered was the ability to plan missions for a drifting reference frame, indicating the need for an autonomous mapping scheme.

## 2. PROBLEM FORMULATION

An AUV is patrolling a given area, by some guidance scheme (e.g. pre-programmed waypoints). The area may contain icebergs, and we are interested in obtaining key parameters about the icebergs in the area (i.e. length, width, and draft). The obtained parameters may be used for example for drift forecasting. The AUV is assumed to be a torpedo-shaped vehicle, more specifically, we assume that the AUV is a REMUS 100 vehicle (Norgren and Skjetne, 2014). The AUV is assumed to have a robust and accurate orientation sensor, capable of measuring the attitude of the AUV at a high rate. Moreover, the AUV is assumed to be equipped with an upwards-looking multibeam echosounder, which will be used throughout this paper as the primary sensor for mapping and decision-making.

No prior information about the iceberg position, or the drift speed and direction is assumed available, but the iceberg geometry is assumed to be convex in the horizontal plane, and the edge of the iceberg is assumed to be steep, and the bottom relatively flat (i.e. approximately constant draft). The drift direction and velocity of the iceberg is assumed unknown and constant. The iceberg can also rotate at a constant rate, independent of the ocean current.

This paper does not consider forward-looking collision avoidance sonars. For practical implementations this will be an important part of the guidance system, to avoid potential impact with a deep drafted iceberg or entrapment.

### 2.1 Guidance system

The guidance system is implemented as a state machine. The initial state of the guidance system is that an iceberg has not been detected, and the AUV will follow a pre-programmed mission plan (e.g. using line-of-sight (LOS) guidance with a waypoint database (Fossen, 2011)). An illustration of the guidance system for iceberg edge-following can be found in Norgren and Skjetne (2015).

Once an iceberg has been detected, the state machine enters the *iceberg detected*-state. This state consist of several sub-states, the initial being a search algorithm for locating the iceberg edge (e.g. an outward spiral search).

Once the edge has been detected, the guidance system enters the *follow edge*-state. The AUV will follow the edge, either until the edge is lost, or until the survey is completed – upon which the state changes back to *iceberg not detected*. If, on the other hand, the edge is lost, the state machine enters a *relocate edge*-state, employing a algorithm for relocating the lost edge.

Timers are used in both the *locate edge*- and the *relocate edge*-state, to avoid getting stuck in these states if the AUV is unable to find the edge. If a certain time passes, the iceberg is deemed lost, and the state changes back to *iceberg not detected*.

### 2.2 Notation

The total time derivative of a variable  $x(t)$  is denoted  $\dot{x}$ . Superscript denotes the reference frame to which a given vector is expressed. For example,  $p^n$  is a position in the NED ( $n$ ), frame. The reference frames used are: NED ( $n$ ), BODY ( $b$ ), and BEAM ( $m$ ). A rotation matrix  $R \in \mathcal{SO}(3)$  between reference frames uses a subscript for the frame transformed from, and superscript for the frame transformed to. For example, rotating from BODY to NED is denoted  $R_b^n$ . Matrices are written in capital letters and vectors are written in small letters. The dimension of each variable will be defined. A variable in the Euclidean space with dimension  $n$  is denoted  $\mathbb{R}^n$ , while matrices of dimension  $n \times m$  are denoted  $\mathbb{R}^{n \times m}$ .  $\mathcal{S}^n$  denotes the  $n$ -dimensional unit sphere.

## 3. GUIDANCE SYSTEM

The overall outline of the guidance system was presented in Section 2.1. The following will present the different algorithms that define the state machine and the state transitions.

### 3.1 Iceberg detection

To transition to the *iceberg detected*-state, the guidance system must determine when an iceberg has been detected. Icebergs often have large drafts, but draft alone may not be enough to distinguish an iceberg from a ridge. Ridges with drafts between 5 meters and 29 meters are reported by Davis and Wadhams (1995), while drafts as deep as 47 meters have been reported by Lyon (1961). For this paper, we will only consider icebergs with principal length larger than 75 meters (medium berg and larger), resulting in a draft larger than 50 meters (McClintock et al., 2007). With this assumption, we can consider an iceberg as detected when the ice draft exceeds 50 meters. A different threshold can be specified depending on the expected iceberg draft, and the expected maximum draft of other ice features. Since acoustic measurements tend to be noisy, at least  $N_{detection} \geq 2$  beams should exceed the specified threshold before an iceberg is considered detected. This is to avoid false detections and, subsequently, to avoid unnecessary time searching for an iceberg that does not exist.

The MBE provides range measurements (the range from the  $i^{\text{th}}$ -beam is denoted  $r_i$ ) between the sonar head and the ensonified object. To extract a point cloud from these measurements (such that e.g. the ice draft is easily identifiable), the measurements are transformed to the *beam*-frame (see Figure 1) by correcting for lever-arm and the AUVs roll and pitch angles.

Let  $R_b^m(\Theta_{nm}) : \mathcal{S}^3 \rightarrow \mathcal{SO}(3)$  denote the rotation matrix with argument  $\Theta_{nm} = [\phi \ \theta]^\top$ , corresponding to the transformation between the  $b$ -frame and the  $m$ -frame:

$$\begin{aligned} R_b^m(\Theta_{nm}) &= R_y(\theta)R_x(\phi) \\ &= \begin{bmatrix} \cos(\theta) & \sin(\theta)\sin(\phi) & \sin(\theta)\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}. \end{aligned} \quad (1)$$

$R_x(\phi)$  is the principal rotation matrix about the x-axis by an angle  $\phi$ , and similarly,  $R_y(\theta)$  is the principal rotation matrix about the y-axis by an angle  $\theta$ . Assuming that the lever-arm from the center of origin of the AUV to the MBE sonar head is  $l_{mbe} = [l_x \ l_y \ l_z]^\top$ , the ranges can be converted to a point cloud in the beam reference frame:

$$p_i^b = l_{mbe} + \begin{bmatrix} 0 \\ r_i \sin(\xi_i) \\ -r_i \cos(\xi_i) \end{bmatrix} = \begin{bmatrix} l_x \\ l_y + r_i \sin(\xi_i) \\ l_z - r_i \cos(\xi_i) \end{bmatrix}, \quad (2)$$

$$p_i^m = \begin{bmatrix} 0 \\ 0 \\ z_{auv}^n \end{bmatrix} + R_b^m p_i^b, \quad (3)$$

where  $p_i^b \in \mathbb{R}^3$  and  $p_i^m \in \mathbb{R}^3$  is the point derived from the  $i^{\text{th}}$  range measurement in the body-frame and the beam-frame, respectively.  $z_{auv}^n$  is the depth of the AUV, assumed measured with a pressure sensor, and  $\phi$  and  $\theta$  is the roll and pitch angles of the AUV, respectively.  $\xi_i$  is the angle of the  $i^{\text{th}}$  beam. These can be measured accurately from high performance inertial measurement units (IMU), typically part the AUV's navigation system.

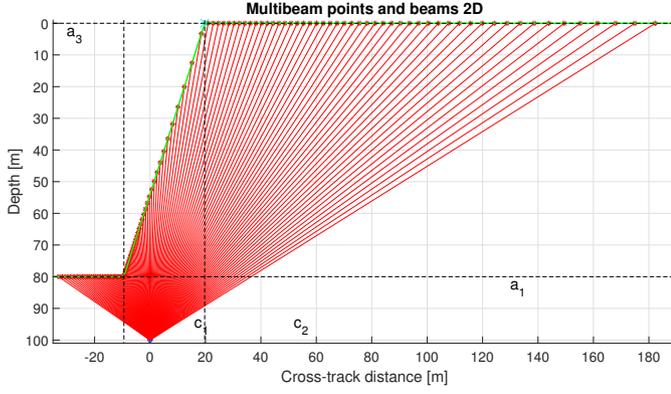


Fig. 1. MBE measurements when running an AUV under an iceberg edge (noise free).

### 3.2 Iceberg edge detection

Once an iceberg candidate has been detected, a mapping of the iceberg must be performed. This is a challenging task, due to the fact that an iceberg will both drift and rotate while the AUV conducts its mapping. By using an upwards-looking Doppler velocity logger (DVL), the relative velocity between the iceberg and the AUV can be measured. However, since rotation does not induce a Doppler shift, the iceberg rotation can not be measured the same way. Mean iceberg translational velocities of 0.2 m/s and maximum velocities of 1 m/s, as well as rotational velocities as high as 90 °/h is reported by Yulmetov and Løset (2014). Thus, a guidance scheme handling a drifting and rotating target of unknown size and shape is needed.

The guidance scheme proposed in this paper aims at circumnavigating the iceberg, by following the iceberg edge. This requires the edge of the iceberg to be identified in real-time using the available measurements.

Assume that the AUV is located near the edge of the iceberg, traversing parallel to the edge. If the walls of the iceberg are near vertical, and the bottom of the draft of the iceberg can be approximated as constant, the range measurements from the MBE will have three clearly identifiable sections; one where the beams reflect off the surface, one where the beams are hitting the near vertical wall, and one where the beams are hitting the bottom of the iceberg. This situation is illustrated in Figure 1.

From Figure 1, the cross-track distance to the edge is clearly identifiable. To effectively extract the cross-track distance to the edge (also in the presence of noise), we propose a simple detection algorithm using piecewise linear regression on all MBE measurements (for more on piecewise linear regression, see (Malash and El-Khaiary, 2010)). That is, we extract three line segments from the points shown in Figure 1. Let  $c_1$  and  $c_2$  be the cross-track distance to the breakpoints between the line segments. The piecewise linear function describing draft as a function of the cross-track distance,  $x$ , can be expressed as

$$F(x, \theta_0) = \begin{cases} f_1(x) = a_1 + b_1x, & x \leq c_1, \\ f_2(x) = a_2 + b_2x, & c_1 < x \leq c_2, \\ f_3(x) = a_3 + b_3x, & x > c_2, \end{cases} \quad (4)$$

where  $b_n$  is the slope of line segment  $n$ , and  $a_n$  is the value where line  $n$  crosses the  $y$ -axis.  $\theta_0$  is a column vector containing all the parameters of (4), i.e all  $a_n$ ,  $b_n$ , and the breakpoints. To be able to include the unknown breakpoints into the regression, (4) can be rewritten to

$$F(x, \theta_0) = \begin{cases} f_1(x) = a_1 + b_1x, & x \leq c_1, \\ f_2(x) = a_1 + c_1(b_1 - b_2) + b_2x, & c_1 < x \leq c_2, \\ f_3(x) = a_1 + c_1(b_1 - b_2) + c_2(b_2 - b_3) + b_3x, & x > c_2. \end{cases} \quad (5)$$

Since we also know that the surface is flat, then  $b_3 = 0$ . Since the iceberg draft is also assumed approximately constant, then  $b_1 = 0$ , and (5) reduces to

$$F(x, \theta) = \begin{cases} f_1(x) = a_1, & x \leq c_1, \\ f_2(x) = a_1 - c_1b_2 + b_2x, & c_1 < x \leq c_2, \\ f_3(x) = a_1 - c_1b_2 + c_2b_2, & x > c_2, \end{cases} \quad (6)$$

with  $\theta = [a_1 \ b_2 \ c_1 \ c_2]$ . Equation (6) can be solved using nonlinear least squares to find the best fit. We also need to enforce the constraint  $c_2 > c_1$ , since  $c_1 \geq c_2$  may cause numerical issues for the nonlinear least squares solver. If we look at (6) and Figure 1, we can see that  $a_1$  will be equal to 80, and  $a_3 = a_1 - c_1b_2 + c_2b_2$  will be equal to 0. The algorithm will also handle the case where the iceberg is on the other side of the AUV, and we will have  $a_1 = 0$  and  $a_3 = 80$ .

We can now formulate a nonlinear optimization problem:

$$\underset{\theta}{\text{minimize}} \ J(x, \theta) = |Z(x) - F(x, \theta)|^2, \quad \text{s.t. } c_2 > c_1, \quad (7)$$

where  $x, Z(x) \in \mathbb{R}^k$ ,  $k = 1, \dots, N_{beams}$  is the cross-track distance and the depth of each beam, respectively. The line between the breakpoints will mark the transition between the iceberg and the surface. By inserting  $x = c_1$  and  $x = c_2$  into the second equation in (6), and choosing the breakpoint corresponding to the minimum draft, that is,  $z_e = \min(F(c_1), F(c_2))$ , then we find the cross-track distance,  $x_{ct}$ , to the surface edge of the iceberg.

To avoid false detections, an edge is only deemed detected if the resulting fit has an RMS error less than some margin (depending on the expected noise and shape deviations from the assumed shape). Thus, all edge-points with RMS error above the threshold are discarded, and counted as a lost edge-point.

Having the cross-track distance and the draft of the iceberg relative to the AUVs position, we can transform the edge-point,  $p_e^n = [x_e, y_e, z_e]^T$ , to the  $n$ -frame using

$$p_e^n = \begin{bmatrix} x_{auv}^n \\ y_{auv}^n \\ 0 \end{bmatrix} + R_z(\psi) \begin{bmatrix} 0 \\ x_{ct} \\ z_{ct} \end{bmatrix} = \begin{bmatrix} x_{auv}^n - \sin(\psi)x_{ct} \\ y_{auv}^n + \cos(\psi)x_{ct} \\ z_{ct} \end{bmatrix}, \quad (8)$$

where  $x_{auv}^n$  and  $y_{auv}^n$  is the position of the AUV in the horizontal plane expressed in the  $n$ -frame, and  $\psi$  is the

heading of the AUV.  $R_z(\psi)$  is the principal rotation matrix about the z-axis by an angle  $\psi$ .

### 3.3 Iceberg edge following

Having detected an iceberg, we want to make the AUV follow its edge. The AUV is only able to measure the cross-track distance to the edge, and no information about the edge in front of the AUV is available. It is also not given that we are able to detect an edge once an iceberg has been detected (for example if the AUV has a heading perpendicular to the iceberg edge). A simple method to search for the iceberg edge is to employ a outward spiral search. If no edge is detected after a certain amount of time, the iceberg is deemed lost, and the AUV continues its survey. On the other hand, if an edge is detected then the bearing of the edge must be calculated for the AUV to be able to follow it.

*Calculating the edge bearing* Since the AUV do not have any information about what is in front of it, the bearing of the iceberg edge is calculated based on the history. This means that all detected edge-points are transformed into the  $n$ -frame, and stored in a queue. The queue has a fixed size, and once full, each new measurement will cause the oldest measurement to be removed from the queue. We require the algorithm to be robust with regards to false edge detections and outliers, and therefore, the bearing of the edge is calculated by using linear regression. Given a queue with  $q$ -elements of edge-points,  $[x_{e,j}^n, y_{e,j}^n]$ ,  $j = 1, \dots, q$ , we define the following relations:

$$S_{XX} = \sum_{j=1}^q (x_{e,j} - \bar{x}_e)^2, \quad (9)$$

$$S_{YY} = \sum_{j=1}^q (y_{e,j} - \bar{y}_e)^2, \quad (10)$$

$$S_{XY} = \sum_{j=1}^q (x_{e,j} - \bar{x}_e)(y_{e,j} - \bar{y}_e), \quad (11)$$

where  $\bar{x}_e$  and  $\bar{y}_e$  represents the average values, that is  $\bar{x}_e = \frac{1}{q} \sum_{j=1}^q x_{e,j}$  and  $\bar{y}_e = \frac{1}{q} \sum_{j=1}^q y_{e,j}$ . Given the general line equation,  $y = \beta_0 + \beta_1 x$  we find the least squares estimates of  $\beta_0$  and  $\beta_1$  from the following (Weisberg, 2014, Chapter 2) assignment:

$$\hat{\beta}_1 = \begin{cases} \frac{S_{XY}}{S_{XX}} & \text{if } S_{XX} \geq S_{YY}, \\ \frac{S_{XY}}{S_{YY}} & \text{otherwise,} \end{cases} \quad (12)$$

$$\hat{\beta}_0 = \begin{cases} \bar{y}_e - \hat{\beta}_1 \bar{x}_e & \text{if } S_{XX} \geq S_{YY}, \\ \bar{x}_e - \hat{\beta}_1 \bar{y}_e & \text{otherwise.} \end{cases} \quad (13)$$

The reason for the two cases in (12) and (13) is that the general line equation does not handle vertical lines ( $S_{XX}$

or  $S_{YY} \rightarrow 0$ ). Note that for the second case, the line equation becomes  $x = \beta_0 + \beta_1 y$ . Now that an estimate of the line for the iceberg edge has been found, the four-quadrant angle for the iceberg edge can be found from:

$$\alpha = \text{atan2}(y_2 - y_1, x_2 - x_1), \quad (14)$$

where the coordinates  $x_1, x_2, y_1$  and  $y_2$  can be found by setting two arbitrary x-coordinates if  $S_{XX} \geq S_{YY}$ , or by setting two arbitrary y-coordinates otherwise, and inserting them into the corresponding line equation.

*Line-of-sight guidance for edge-following* In the previous section, we calculated the bearing of the edge. In this section we will modify the ordinary LOS guidance scheme (Fossen, 2011, Chapter 10) to be able to converge to and follow a path parallel to the iceberg edge. For a lookahead-based steering scheme, the course angle assignment can be divided into two parts

$$\chi_d(e) = \chi_p + \chi_r(e), \quad (15)$$

where  $\chi_p$  is the path-tangential angle, given by the edge bearing  $\chi_p = \alpha$ . To find the velocity-path relative angle  $\chi_r(e)$  we need the cross-track error  $e$  to the desired path,

$$e = -(x_{ct} + s\epsilon), \quad (16)$$

where  $\epsilon$  is used to offset the AUVs trajectory towards the estimated center of the iceberg, and  $s \in [-1, 1]$  shifts the offset, depending on if the iceberg is to the left ( $s = -1$ ) or to the right ( $s = 1$ ) of the AUV. This is to make sure we map as much as possible of the iceberg, and to always have measurements of the AUV - iceberg relative speed from DVL measurements. Since the swath-width of the MBE will vary with the distance from the AUV to the iceberg, the offset should ideally have been calculated from this distance. This is outside the scope of this paper.

The velocity-path relative angle can now be found from the ordinary lookahead-based steering law:

$$\chi_r(e) = \arctan \left( -K_p e - K_i \int_0^t e(\tau) d\tau \right), \quad (17)$$

with proportional and integral gain given as  $K_p$  and  $K_i$ , respectively. The integral effect in (17) is necessary to guarantee convergence to the desired path in the presence of disturbances, like ocean currents. The full guidance law can be written as:

$$\chi_d(e) = \alpha + \arctan \left( -K_p e - K_i \int_0^t e(\tau) d\tau \right). \quad (18)$$

*Relocating a lost edge* The guidance scheme shown in (18) will only work as long as it is possible to detect the edge of the iceberg. If a corner of the iceberg causes the edge to change with more than 90 degrees, then the AUV will not be able to detect the iceberg any longer. Overshoot

in the AUV’s path-following control function, e.g. due to a sharp corner, may also result in loss of edge detection. Thus, we need to handle the scenario where an edge is lost.

The edge is deemed lost if the algorithm shown in Section 3.2 is unable to detect an edge for a given number of iterations. A counter is used to count the number of edges not detected, and if this counter reaches a given threshold ( $N_{not\ Detected}$ ), the state machine switches to the relocate edge-state. If this happens, the queue containing all the detected edge-points will be cleared. By assuming that the iceberg is convex and that the AUV has followed the desired track, we know that the AUV cannot have passed in under the iceberg. Therefore, to relocate the edge, the AUV will initiate a turn in the direction the iceberg was detected until the iceberg edge is relocated. Once the edge is detected again, the edge-following algorithm takes over.

One could argue that a more sophisticated commencement of the edge-following could be employed. For example, once the edge has been detected again, the AUV could circle around to the point where the edge was lost, entering this point heading in the direction of the newly detected edge. This way, we would be able to map the whole edge. This is left as further work.

### 3.4 Termination criterion

When the iceberg has been circumnavigated, the AUV should end the edge-following mode, and continue its pre-programmed mission. A method for switching between waypoints in the regular LOS algorithm, called *circle of acceptance* can also be used in this context (Fossen, 2011, Chapter 10). That is, the survey is considered completed when the position of the AUV in the horizontal plane  $[x_{auv}, y_{auv}]$  satisfies:

$$[x_s - x_{auv}]^2 + [y_s - y_{auv}]^2 \leq R^2, \quad (19)$$

where  $[x_s, y_s]$  is the start position of the survey and  $R$  is the radius of the circle of acceptance. The start position of the survey can either be chosen as the position of the first detected edge point (offset by  $\epsilon$  to lie on the desired path of the AUV), or it can be chosen as the position of the AUV, once converged to the desired path. When choosing  $R$ , the estimated duration of the survey and the drift of the iceberg must be taken into account, or the AUV may miss on the circle after a complete circumnavigation of the iceberg. One solution is to make  $R$  grow according to the estimated iceberg drift rate as time passes.

## 4. AUV SIMULATOR

To perform a simulation study of the developed guidance algorithm, a numerical model describing the AUV dynamics, the environmental loads (ocean currents), and a drifting iceberg is convenient. Last, a model of an MBE sensor is needed to emulate range measurements. This section will describe the models, and outline the control system for the AUV.

### 4.1 AUV model

To analyze the behavior of the guidance algorithm under realistic AUV motions, a 6 degree-of-freedom (DOF)

model is used. The measurements from the MBE sensor are greatly affected by AUV motions like rolling and pitching, since the sonar head is fixed to the the AUV body. The targeted AUV during the simulator development is the REMUS 100 AUV (Norgren and Skjetne, 2014). The model parameters have been taken from a previous REMUS 100 model by Prestero (1994).

The general 6 DOF equations of motion of a marine craft can be written on vectorial form (Fossen, 2011)

$$\dot{\eta} = J_{\Theta}(\eta)\nu, \quad (20a)$$

$$M\nu_r + C(\nu_r)\nu_r + D(\nu_r)\nu_r + g(\eta) = \tau + \tau_{env}, \quad (20b)$$

where  $\eta = [p_{auv}^n \ \Theta_{nb}]^T$  is the position and orientation vector of the AUV.  $p_{auv}^n \in \mathbb{R}^3$  denotes the position of the AUV in the  $n$ -frame, while  $\Theta_{nb} \in \mathcal{S}^3$  a vector of Euler angles.  $\nu \in \mathbb{R}^6$  is the linear and angular velocities of the AUV, expressed in the  $b$ -frame, and  $\tau, \tau_{env} \in \mathbb{R}^6$  is the forces and moments acting on the AUV in the body-fixed frame from the control surfaces and the environmental loads, respectively.  $\nu_r = \nu - \nu_c \in \mathbb{R}^6$  is the relative velocity vector, and  $\nu_c$  is the velocity vector of the ocean currents. In the model we have assumed irrotational ocean currents.

Equation (20b) defines the kinetics of the AUV, while the velocity transformation from the  $b$ -frame to the  $n$ -frame is expressed in (20a) (Fossen, 2011).  $M = M_{RB} + M_A \in \mathbb{R}^{6 \times 6}$  is the rigid-body inertia and added mass of the AUV, while the centripetal and Coriolis rigid-body and added mass are denoted  $C(\nu) = C_{RB}(\nu) + C_A(\nu_r) \in \mathbb{R}^{6 \times 6}$ .  $D(\nu_r) \in \mathbb{R}^{6 \times 6}$  represents hydrodynamic damping, and restoring forces are given by  $g(\eta) \in \mathbb{R}^6$ . The coefficients for these matrices can be found in (Prestero, 1994).

### 4.2 Environment

As an AUV generally is submerged for most of the mission, neither wind nor waves have been considered. The only environmental load is the ocean current, which is considered constant and irrotational. It is also assumed that there is no vertical component from the ocean current. The resulting ocean current velocity vector, expressed in the  $b$ -frame, is

$$\begin{aligned} \nu_c &= [u_c \ v_c \ 0 \ 0 \ 0 \ 0]^T \\ &= [V_c \cos(\psi_c) \ V_c \sin(\psi_c) \ 0 \ 0 \ 0]^T, \end{aligned} \quad (21)$$

where  $\psi_c = \xi_c - \psi$ , and  $\xi_c$  is the ocean current direction relative to the  $n$ -frame, and  $\psi$  is the heading of the AUV.  $V_c$  denotes the speed of the ocean current. The linear drift of the iceberg is assumed to be equal to the ocean current drift speed and direction, but the rotational velocity of the iceberg can be set to a nonzero value independently.

### 4.3 Control surfaces

The pitch of the REMUS AUV is controlled by two horizontal fins, and similarly, the heading of the AUV is controlled by two vertical fins. Both the stern planes and the rudder planes are moving together; that is, the two horizontal fins cannot be controlled individually, and

neither can the vertical fins. Prestero (1994) derived the following expressions for the forces and moments from the rudder and pitch fins

$$Y_r = \frac{1}{2} \rho c_{L,\alpha} S_{fin} [u^2 \delta_r - uv - x_{fin}(ur)], \quad (22)$$

$$Z_s = \frac{1}{2} \rho c_{L,\alpha} S_{fin} [u^2 \delta_s + uv - x_{fin}(uq)], \quad (23)$$

$$M_s = \frac{1}{2} \rho c_{L,\alpha} S_{fin} x_{fin} [u^2 \delta_s + uv - x_{fin}(uq)], \quad (24)$$

$$N_r = \frac{1}{2} \rho c_{L,\alpha} S_{fin} x_{fin} [u^2 \delta_r - uv - x_{fin}(ur)], \quad (25)$$

where  $\rho$  is the density of water,  $S_{fin}$  is the fin area,  $x_{fin}$  is the x-coordinate of the fin placement (given in the  $b$ -frame), and  $c_{L,\alpha}$  is the fin lift coefficient estimated from an empirical formula taking the angle of attack,  $\alpha$ , as argument. The remaining variables of (22)-(25) are the linear and angular velocities of the AUV,  $\nu = [u \ v \ w \ p \ q \ r]^\top$ , and  $\delta_s$  and  $\delta_r$  are the stern and rudder angles relative to the body x-axis, respectively. The coefficients are found in (Prestero, 1994).

The propeller model of Prestero (1994) was designed for small deviations from a given forward speed. Since the control system of the REMUS vehicle outputs a desired propeller shaft speed, we chose to alter the propeller shaft model by adopting the model of Carlton (2012):

$$X_p = K_T \rho D^4 n^2, \quad (26)$$

$$K_p = K_Q \rho D^5 n^2, \quad (27)$$

where  $K_T$  and  $K_Q$  are the strictly positive thrust and torque coefficients,  $D$  is the propeller diameter, and  $n$  is the shaft speed given in revolutions-per-second.  $K_T$  and  $K_Q$  has been calculated from the REMUS propeller characteristics by Allen et al. (2000). Setting  $\tau = [X_p \ Y_r \ Z_s \ K_p \ M_s \ N_r]^\top$  concludes the AUV model.

#### 4.4 Beam range simulator

The beam range simulator used in conjunction with the AUV dynamics has been developed by Holsen (2014), and a description of the beam simulator will be given in the following. The vector describing the direction of each beam is given by

$$\frac{r_{MBE,i}^n}{|r_{MBE,i}^n|} = R_b^n(\Theta_{nb}) R_{MBE,i}^b(\Theta_{bMBE,i}) [0 \ 0 \ -1]^\top, \quad (28)$$

where  $r_{MBE,i}^n \in \mathbb{R}^3$  is the position vector of the  $i^{\text{th}}$  beam.  $\Theta_{bMBE,i} \in \mathcal{S}^2$  is the angles of the beams expressed relative to the  $b$ -frame, and  $R_{MBE,i}^b(\Theta_{bMBE,i}) = R_y(\theta_{MBE,i}) R_x(\phi_{MBE,i})$  is the rotation matrix between the  $i^{\text{th}}$  beam and the  $b$ -frame.

The range of each beam is estimated through an iterative procedure. Let  $e_i$  be the current estimate of the  $i^{\text{th}}$  beam range. We calculate the beam-end position,  $p_{MBE,i}^n = [x_{MBE,i} \ y_{MBE,i} \ z_{MBE,i}]^\top$  from

$$p_{MBE,i}^n = p_{AUV}^n + \frac{r_{MBE,i}^n}{|r_{MBE,i}^n|} e_i. \quad (29)$$

The surface, which in this case represents the water surface or the iceberg, is represented as a 3D digital terrain map, where each entry in the matrix has a north-, east-, and depth-coordinate. To get the depth at a given position, bilinear interpolation between the four nearest neighbors is used, resulting in a depth at a given position,  $Z_{surface}(x, y)$ . The error between the beam-end position and the surface is

$$e_{Z,MBE} = z_{MBE,i} - Z_{surface}(x_{MBE,i}, y_{MBE,i}). \quad (30)$$

If the error is positive, the beam range is too small, while a negative error tells us that the beam range is too large. The beam range is increased by an increment until the error becomes negative, then binary search is used to reduce the error to a sufficient accuracy. Binary search could have been used for the whole search; however, this caused the beams to hit the wrong surface (beams hitting the water surface when they should have been hitting the iceberg) under some conditions.

#### 4.5 AUV control system

To implement a guidance algorithm, a control system is necessary for the AUV to be able to follow the commands given. The control system is divided into: speed control, depth control, and heading control. Only a brief overview of the control system will be given here. A speed controller has not been implemented, and the propeller has a constant shaft speed through all simulations.

The depth controller is implemented as two loops: the first loop is a proportional-integral (PI) controller on the depth error that generates a pitch setpoint, while the second loop is a proportional-integral-derivative (PID) controller that controls the stern planes of the AUV. The heading controller is a PID controller that controls the rudder fins of the AUV. All setpoints passed to the controllers are passed through reference models to avoid excessive changes in control commands.

## 5. SIMULATION RESULTS

The AUV simulator presented in Section 4 is implemented in MATLAB<sup>®</sup> and Simulink<sup>®</sup>. The beam range simulator presented in Section 4.4 is implemented in C++ and compiled as a Simulink S-function for efficiency. Similarly, the guidance system presented in Section 3 is also implemented as a C++ S-function. The guidance algorithm state machine is implemented using the Boost Statechart library (Boost, 2015), and the Eigen C++ library is used for linear algebra and matrix and vector representation (Guennebaud, 2015). For solving the nonlinear least square optimization problem in the edge-detection algorithm presented in Section 3.2, the ALGLIB library is used (Bochkanov, 2015).

A simulation of the complete guidance system can be seen in Figure 2. In this simulation the iceberg is stationary and ideal MBE measurements are assumed. The shape

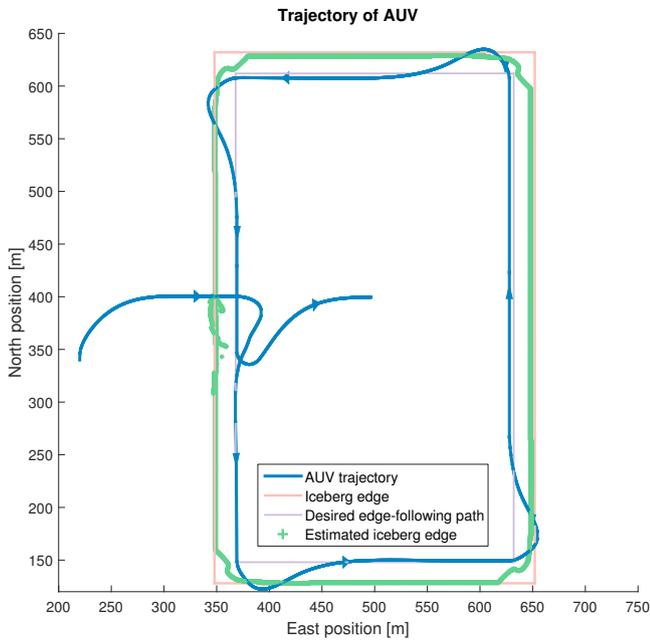


Fig. 2. AUV trajectory around a stationary, simplified iceberg with steep walls.

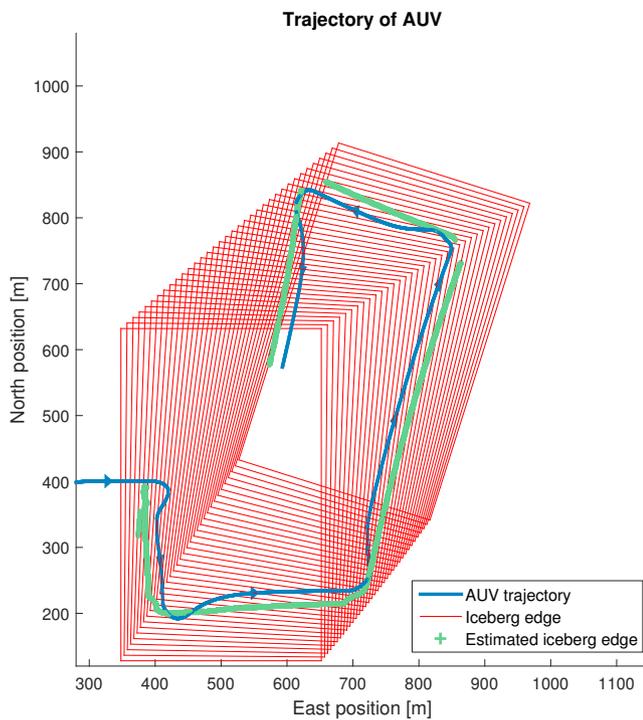


Fig. 3. AUV trajectory around a simplified iceberg with steep walls, drifting with translational velocity of 0.5 m/s and rotational velocity of 90 deg/h.

of the iceberg used in this simulation is a rectangular iceberg, with slightly sloped walls. This very simplified shape is used to test the guidance algorithm under ideal conditions. As can be seen from Figure 2 the AUV is able to circumnavigate the iceberg, and we can see that the detected edge corresponds to the actual edge quite well, except at the corners. There are also some edge detection anomalies in the area where the iceberg was first detected. This is caused by the large heading change of the AUV

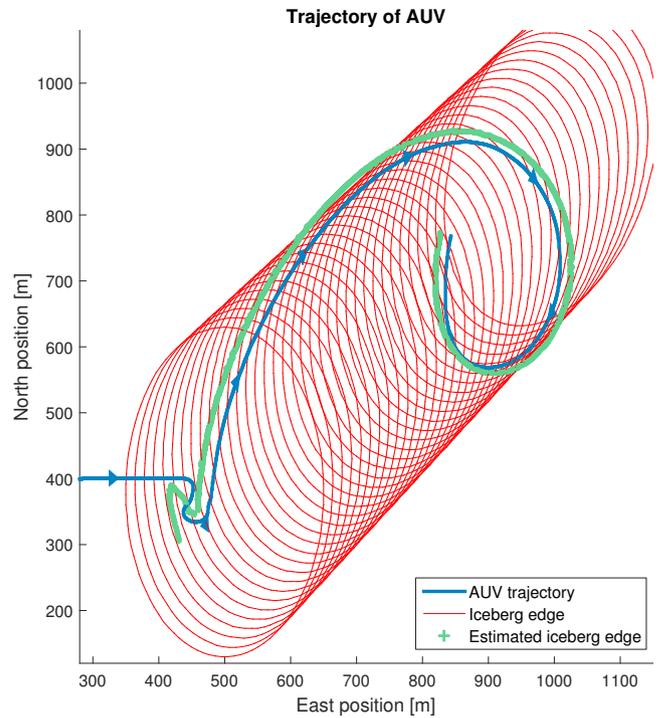


Fig. 4. AUV trajectory around a rounded, oval iceberg drifting with translational velocity of 1.0 m/s and rotational velocity of 90 deg/h.

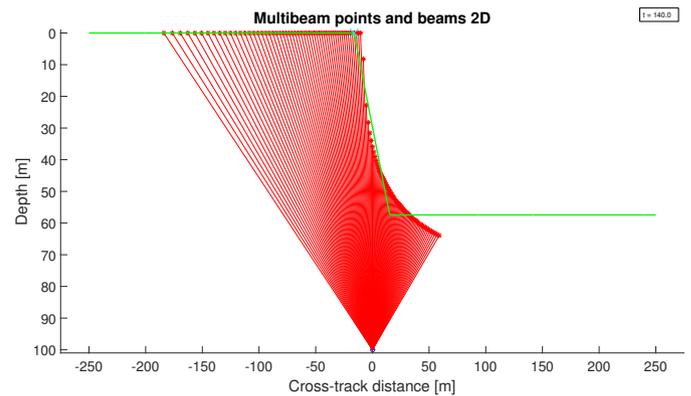


Fig. 5. A detected edge under the rounded, oval iceberg.

giving some inaccurate detections. There are some steady-state offsets between the detected and the real edge, which stem from the beams of the MBE not hitting the surface exactly at the edge.

Figure 3 shows the same simple iceberg, but in this case the iceberg is drifting with a constant velocity of 0.5 m/s and rotating with a rate of 90 °/h. The AUV is able to circumnavigate the iceberg, and we can easily see the warping of the path caused by the translation and rotation of the iceberg.

A translational velocity higher than 0.5 m/s caused problems at the corners of the iceberg. The edge was lost, and the relocate edge algorithm was invoked. However, due to the large velocity of the iceberg, the AUV missed the next iceberg side, but detected the side that was already mapped. This result indicate the need for a more sophisticated relocate edge algorithm.

Figure 4 shows a simulation tracking the edge of a rounded, oval iceberg, of similar length, width, and draft as the simple iceberg. This is a more realistic shape than the shape previously tested, and the shape does not conform to our assumptions of steep walls and flat bottom. From Figure 4 we can, however, see a successful circumnavigation of the rounded iceberg drifting with a velocity of 1 m/s and rotating with a rate of 90 °/h. We can see a U-turn in the start of the mapping, which is caused by a lost edge and initiation of the relocate edge algorithm. One run of the edge detection algorithm is presented in Figure 5.

## 6. CONCLUSION

Throughout this paper, a guidance algorithm for circumnavigating an iceberg has been detailed. A state machine is used for transitioning between the different situations handled by the guidance scheme. When no iceberg is detected, the AUV follows a pre-programmed mission plan, and once an iceberg is detected, the AUV starts identifying the edge. The edge is detected using piecewise linear regression, and the bearing of the edge is calculated using linear regression on several edge-points. Using the calculated bearing and cross-track error to the edge, a LOS-based scheme is used to follow the iceberg edge. A simulation study shows that the guidance algorithm successfully guides the AUV around the iceberg, under the assumption of noise free MBE range measurements, and noise free position and attitude measurements. The guidance algorithm is also shown to successfully handle nonzero linear drift of the iceberg, as well as nonzero drift and rotation of the iceberg. The algorithm is also shown to handle shapes that does not strictly conform to the assumptions of flat iceberg bottom and steep walls.

Further work includes testing the algorithm with noisy range measurements, as well as noisy position and attitude measurements. Furthermore, the edge detection algorithm should be improved to robustly handle a generic iceberg shape, and to avoid potential problems with nonlinear least squares optimization (i.e. local minima). Improved methods for tracking the initial position of the edge-following procedure, and determine when complete circumnavigation has been conducted, will be investigated. A more sophisticated relocate edge search algorithm will also be investigated.

## ACKNOWLEDGEMENTS

This work was supported by the Research Council of Norway (RCN) through the Centre of Excellence AMOS (RCN-project 223254), and the Centre for Research-based Innovation SAMCoT (RCN-project 203471).

## REFERENCES

- Allen, B., Vorus, W., and Prestero, T. (2000). Propulsion system performance enhancements on REMUS AUVs. In *OCEANS*, 1869–1873. Providence, RI, USA.
- Bochkanov, S. (2015). ALGLIB. [www.alglib.net](http://www.alglib.net).
- Boost (2015). The Boost Statechart Library. [http://www.boost.org/doc/libs/1\\_57\\_0/libs/statechart/](http://www.boost.org/doc/libs/1_57_0/libs/statechart/).
- Carlton, J.S. (2012). *Marine Propellers and Propulsion*. Butterworth-Heinemann, Oxford, UK, third edition.
- Davis, N.R. and Wadhams, P. (1995). A statistical analysis of Arctic pressure ridge morphology. *Journal of Geophysical Research*, 100(C6), 915–925.
- Eik, K. (2008). Review of Experiences within Ice and Iceberg Management. *Journal of Navigation*, 61(04), 557–572.
- Forrest, A.L., Hamilton, A.K., Schmidt, V., Laval, B.E., Mueller, D., Crawford, A., Brucker, S., and Hamilton, T. (2012). Digital terrain mapping of Petermann Ice Island fragments in the Canadian High Arctic. In *Proceedings of the 21st IAHR International Symposium on Ice*, 710–721. Dalian, China.
- Fossen, T.I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Ltd.
- Francois, R. and Nodland, W. (1972). Unmanned Arctic research submersible (UARS) system development and test report. Technical Report No. APL-UW 7219, University of Washington.
- Guennebaud, G. (2015). Eigen: a C++ linear algebra library. <http://eigen.tuxfamily.org/>.
- Holsen, S.A. (2014). Evaluation of Software Platforms and Development of Terrain Relative Navigation for UUV. Project thesis, NTNU.
- Lyon, W. (1961). Ocean and sea-ice research in the Arctic Ocean via submarine. *Transactions of the New York Academy of Sciences*, 23(8), 662–674.
- Malash, G.F. and El-Khaiary, M.I. (2010). Piecewise linear regression: A statistical method for the analysis of experimental adsorption data by the intraparticle-diffusion models. *Chemical Engineering Journal*, 163, 256–263.
- McClintock, J., McKenna, R., and Woodworth-Lynas, C. (2007). Grand Banks iceberg management. PERD/CHC Report 20-84, Report prepared by AMEC Earth & Environmental, St. John's, NL, Canada, R.F. McKenna & Associates, Wakefield, QC, Canada, and Petra International Ltd., Cupids, NL, Canada.
- Norgren, P. and Skjetne, R. (2014). Using Autonomous Underwater Vehicles as Sensor Platforms for Ice-Monitoring. *Modeling, Identification and Control*, 35(4), 263–277.
- Norgren, P. and Skjetne, R. (2015). Iceberg detection and edge-following using AUV with multibeam sonar. In *Proceedings of the 23rd International Conference on Port and Ocean Engineering under Arctic Conditions*, 1–13. Trondheim, Norway.
- Prestero, T. (1994). *Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle*. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.
- Wadhams, P., Wilkinson, J.P., and Kaletsky, A. (2004). Sidescan sonar imagery of the winter marginal ice zone obtained from an AUV. *Journal of Atmospheric and Oceanic Technology*, 21(9), 1462–1470.
- Wadhams, P., Wilkinson, J.P., and McPhail, S.D. (2006). A new view of the underside of Arctic sea-ice. *Geophysical Research Letters*, 33(4), L04501.
- Weisberg, S. (2014). *Applied linear regression*. John Wiley & Sons Ltd.
- Yulmetov, R. and Løset, S. (2014). Kinematic Characteristics of Sea Ice and Iceberg Drift in the Greenland Sea. In *Proceedings of the 22nd IAHR International Symposium on Ice*, 727–734.