Proceedings of the ASME 2020 39th International
Conference on Ocean, Offshore and Arctic Engineering
OMAE2020
August 3-7, 2020, Virtual, Online

OMAE2020-19117

GUIDANCE MODEL REPRESENTING AN ICE FIELD FOR PATH-PLANNING IN ICE MANAGEMENT

Jon Bjørnø *
Department of Marine Technology
Norwegian University of Science
and Technology (NTNU)
Sustainable Arctic Marine and
Coastal Technology (SAMCoT)
NO-7491 Trondheim, Norway
Email: jon.bjorno@ntnu.no

Mathias Marley
Department of Marine Technology
Norwegian University of Science
and Technology (NTNU)
Centre for Autonomous Marine
Operations and Systems (AMOS)
NO-7491 Trondheim, Norway
Email: mathias.marley@ntnu.no

Roger Skjetne
Department of Marine Technology
Norwegian University of Science
and Technology (NTNU)
Sustainable Arctic Marine and
Coastal Technology (SAMCoT)
Centre for Autonomous Marine
Operations and Systems (AMOS)
NO-7491 Trondheim, Norway
Email: roger.skjetne@ntnu.no

ABSTRACT

In the work presented in this paper, the problem on how to represent a simplified ice field in a guidance model, enabling path and maneuver planning for IM operation, has been studied. The use of B-splines and other basis functions are considered to represent relevant guidance information over the 2D drifting ice field. A weight value is computed and updated at locations that represents broken ice (visited by an icebreaker) versus unbroken ice. The guidance model will ensure that there is a continuous representation of the state of the ice field during the operations. The drifting behavior of the ice field is incorporated into the guidance model. The model will be updated with new (solid) ice that is formed at the beginning of the ice field, and it will continuously be updated in the path where the icebreaker moves. To simulate the maneuvers of the icebreaker, a dynamic model is used, and the ice breaking effect where the ice field is continuously broken into smaller ice floes is included in the model. This representation of an ice field can be used in a path-planning algorithm to determine the icebreaker path in a moving ice environment in order to reduce the ice field into small enough ice floes and reduce the load on the protected structure.

*Address all correspondence to this author.

Keywords: Ice Management, B-splines, Dynamic environment, Ice breaking, Ice field representation.

INTRODUCTION

The Arctic is an important area with an increasing interest involving marine activities over the past years. Some reasons for this are the presence of rich natural resources and its strategically important location including, e.g., the Northern Sea Routes. The rough climate, and especially presence of sea ice, causes high complexity for both marine operations and design of offshore structures, compared to open water activities. While open water activities have a substantial empirical data foundation for estimating risk and loads on offshore structures, Arctic marine designs suffer from large uncertainties with limited empirical basis. To compensate for the uncertainties and reduce the risk of operations in the Arctic, ice management (IM) is often deployed (e.g., Molikpaq and Kulluk in Canadian Arctic waters).

The main purpose of IM is to reduce the ice loads downstream to ensure protection of offshore structures. This is typically done by using ice-breaking vessels to reduce the incoming ice into less severe ice features that will not damage the protected structure. Figure 1 illustrates a possible future Arctic drilling op-

FIGURE 1: Illustration of a possible future Arctic drilling operation. Illustration: Bjarne Stenberg. Copyright: NTNU.

eration: an Arctic drillship on dynamic positioning (DP), aided by icebreakers, and having unmanned underwater vehicles, various drones, and satellites for ice surveillance. IM is a complex and integrated system that involves detection, tracking, forecasting, decision making, and ice mitigation typically through breaking/fracturing threatening ice features [1]. When planning for ice operations in the Arctic, several decisions must be made. For example, how many vessels that will be necessary and what is the most efficient pattern for managing the ice under the given circumstances, are important to decide. Simplified models, such as a kinematic model [2, 3, 4], can be used as a decision-making tool.

Due to a limitation in ice surveillance and a lack of reliable physically based models to efficiently characterize the ice-structure interaction processes at operational scale [5], the IM operations today are largely based on operational experiences with limited quantifiable criteria or guidance information available. However, technological advancements are improving to more accurately retrieve important ice characteristics in real time and to establish physically based models that efficiently can predict the ice-structure interaction processes at full scale; see for instance [6, 7, 8, 9].

A motion control system of marine crafts typically consists of a guidance system, navigation system, and control system where the task of the guidance system is to generate desired position, velocity, and acceleration for use in the motion control system [10]. Jørgensen and Skjetne developed an online estimation design for a drifting two dimensional ice topography [11] and later for three dimension [12]. Motivated by the topography model in [12], we propose in this paper a guidance model to represent the drifting, partly broken ice cover, to enable path and maneuver planning for IM operation, by the use of B-splines.

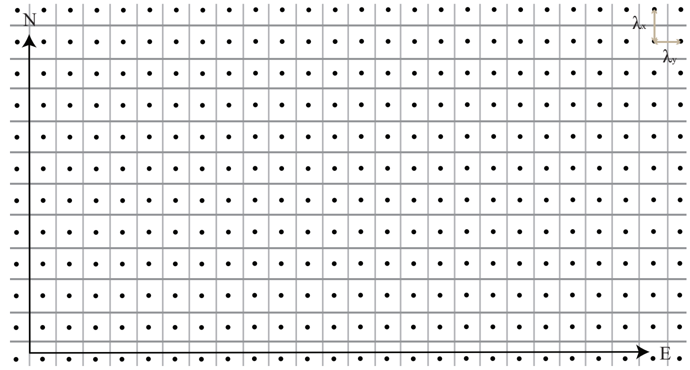

FIGURE 2: Illustration on how the operation area for the icebreaker is discretized into nodes and cells.

PROBLEM FORMULATION

The use of B-splines or other basis functions will be considered to represent relevant guidance information over a 2D drifting ice field. A weight value must be computed and updated for each location that has been visited by the icebreaker to represent broken ice versus unbroken ice. The weight can be binary, representing visited or not visited at that part of the ice cover, or it may take a more continuous value to, e.g., represent the breakability of the ice cover [6]. The guidance model must ensure that there is a continuous representation of the state of the ice field during the operations. The drift dynamics of the ice field shall be incorporated into the guidance model. The model must be updated with the new (solid) ice that is formed at the beginning of the ice field, and it must be continuously updated where the icebreaker moves.

An operation area is discretized in x- and y-direction by cells, as illustrated in Figure 2, where each cell will be updated by the icebreaking status within that cell. Each cell will be represented by a central node that holds the guidance state of each cell. The size $\lambda_x \times \lambda_y$ of the cells are set equal to the breadth of the icebreaker used. However, the cell size can be both increased or decreased according to the accuracy needed, but finer resolution makes the computational time of the B-splines longer.

B-splines

The B-splines are a set of piecewise polynomial functions put together to create a suitable basis for the vector space of the splines. B-splines were introduced in [13, 14] and later in [15].

According to [16] the B-spline of degree d and order $d+1$ with knots $k_i < ... < k_{i+d+1}$ is defined recursively as

$$\phi_i^{d+1}(x) = \frac{x - k_i}{k_{i+d} - k_i}\phi_i^d(x) + \frac{k_{i+d+1} - x}{k_{i+d+1} - k_{i+1}}\phi_{i+1}^d(x), \quad (1)$$

where the initial B-spline of order 1 is defined as

$$\phi_i^1(x) = \begin{cases} 1, \text{ if } x \in [k_i, k_{i+1}] \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

Equations (1) and (2) is only one way of defining B-splines, called *Cox de Boor recursion formula*. This representation of B-splines is that the B-splines themselves are splines.

The B-spline basis function comes with a set of basic properties, extensively studied [17, 18]. These properties are:

Positivity: The B-splines is non-negative, i.e., $\phi_i^d(x) \geq 0$ for all $x \in \mathbb{R}$.
Local support: The support of B-splines is bounded, i.e., it is only larger than zero over its natural definition domain: $\phi_i^d(x) = 0$ if $x \notin [k_i, k_{i+d+1}]$.
Boundary values: The boundary values are zero, i.e., $\phi_i^{d+1}(k_i) = \phi_i^{d+1}(k_{i+d+1}) = 0$.
Continuity: For a given degree d, the B-splines belongs to the highest possible class of continuity for a piecewise polynomial function, i.e., $\phi_i^d \in \mathtt{C}^{d-2}([k_i, k_{i+d+1}])$.
Derivatives: The derivative of a B-spline of degree d is a linear combination of B-splines of degree $d-1$, i.e., $\frac{d}{dx}\phi_i^d(x) = \frac{d-1}{k_{i+d-1}-k_i}\phi_i^{d-1}(x) - \frac{d-1}{k_{i+d}-k_{i+1}}\phi_{i+1}^{d-1}(x)$.

Linear combination of B-splines

A single B-spline of degree d will span over $d+1$ knot intervals. Then $n-d$ independent B-splines can be defined from $\mathbf{k} = k_0 \leq \ldots \leq k_n$. Since the vector space $\mathtt{S}_d(k_0, \ldots, k_n)$ now has $n+d$ dimensions, there is a need for $2d$ supplemental B-splines to form a basis of $\mathtt{S}_d(k_0, \ldots, k_n)$. These B-splines are defined by adding $2d$ extra knots to the knot sequence, to satisfy

$$\begin{gathered} k_d \leq \ldots \leq k_{-1} \leq k_0, \\ \text{and } k_n \leq k_{n+1} \leq \ldots \leq k_{n+d}. \end{gathered} \tag{3}$$

These extra knots are called the *boundary knots*.

A B-spline representation, z, is a linear combination of the $n+d$ basis functions ϕ_i^{d+1} for $i \in [-d, n-1]$, that is

$$z(x) = \sum_{i=-d}^{n-1} c_i \phi_i^{d+1}(x), \tag{4}$$

where c_i are coefficients defining the weight of each B-spline; see [16] for more details.

de Boor's algorithm Using Cox de Boor recursion formula we can now describe the de Boor's algorithm. The de Boor's algorithm evaluates $z(x)$ through a recursion formula instead of computing the B-spline functions $\phi_i^d(x)$. This recursion formula is according to [17, 19] given by

$$b_i^r = (1-\alpha_i^r) d_{i-1}^{r-1} + \alpha_i^r d_i^{r-1}, \tag{5}$$

$$\alpha_i^r = \frac{x - k_i}{k_{i+1+d-r} - k - i}, \tag{6}$$

where b_i^r are the new control points with $b_i^0 := c_i$ for $i = n-d, \ldots, n$, and $r = 1, \ldots, d$ is the number of iterations. Once the iterations are complete the desired result becomes $z(x) = b_n^d$.

The de Boor's algorithm is a more efficient way to calculate the B-splines than explicit calculation using the Cox de Boor recursion formula, since it does not compute terms which are guaranteed to be multiplied by zero.

Optimal de Boor's algorithm for computers The de Boor's algorithm presented in (5) and (6) is not optimized for implementation in a computer since it requires memory for $(d+1)+d+\ldots+1 = (d+1)(d+2)/2$ temporary control points b_i^r. This means that each temporary control point is read twice and written once. To solve this, the iteration order can be reversed by counting down instead of up. This will then only require memory for $p+1$ temporary control points since b_i^r is reusing the memory for b_i^{r-1}. The recursion formula is then given by

$$b_j^r = (1-\alpha_j^r) b_{j-1}^{r-1} + \alpha_j^r b_j^{r-1}; \quad j = d, \ldots, r, \tag{7}$$

$$\alpha_j^r = \frac{x - k_{j+n-d}}{k_{j+1+d-r} - k_{j+n-d}}, \tag{8}$$

where $b_j^r := c_{j+n-d}$ for $j = 0, \ldots, p$, iterated for $r = 1, \ldots, d$ and j must be counted down in (7). The result is $z(x) = b_d$ when the iterations are complete.

B-Splines in higher dimensions

We can use the linear combination to extend the B-splines to a higher dimension. According to [16] this can be done by different approaches, and the most simple one is to extend the univariate B-splines to two or more variables. Since we only need two variables to represent a guidance model over the ice field, we limit the B-splines to this.

Let $\{k_{-d_x}, \ldots, k_{n_x+d_x}\}$ and $\{l_{-d_y}, \ldots, l_{n_y+d_y}\}$ be two knot sequences. The product B-spline of degree d_x in x-direction and d_y in y-direction is the map $z : [k_0; k_{n_x}] \times [l_0; l_{n_y}] \to \mathbb{R}$, defined as

$$z(x,y) = \sum_{i=-d_x}^{n_x-1} \sum_{j=-d_y}^{n_y-1} c_{ij} \phi_i^{d_x+1}(x) \phi_j^{d_y+1}(y), \tag{9}$$

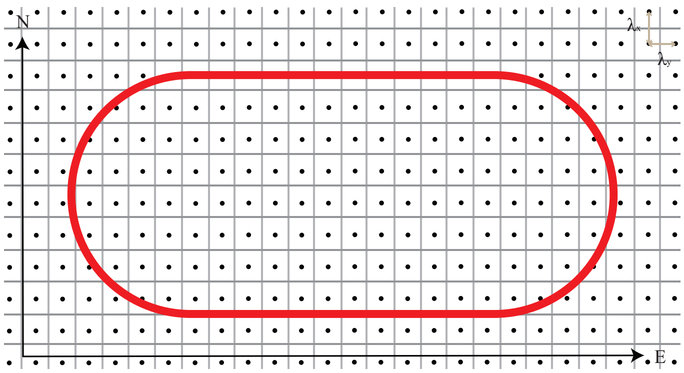

FIGURE 3: Illustration of the racetrack-shaped icebreaker path used.

where ϕ_i^{d+1} is defined in (1). Combining the de Boor's algorithm and (9), the product for B-splines with two variables becomes

$$z(x,y) = b_{d_x} b_{d_y}. \tag{10}$$

Moving ice field representation

To represent the simplified ice field, the B-spline equations for 2D, equations (9) and (10), are used. This will then represent an ice field with level ice that will be broken up by an icebreaker. The level ice will initially have the weight 0 in all nodes, indicating unbroken ice. Later, different values between 0 and -1 will be computed as the simulations unfolds. When the icebreaker has broken up the ice at given location, it sets the weight of the cells visited to -1. The area between the nodes becomes interpolated.

Since the B-spline representation of the ice field is stationary, it is necessary to apply dynamics such that it can simulate a moving surface. This is done by letting the position, $s = (x,y)$, represent the dynamics of the ice field. In this case the ice field has a constant drift velocity, $\dot{s} = v = \mathrm{col}(v_x, v_y)$, which will give the position s for each discretization node to be $s(t) = s(t - \Delta t) + v\Delta t$.

For each time step the icebreaker position is checked against the moving discretization nodes of the ice field. If the icebreaker position is within a given cell the corresponding node value is set to -1, and a new calculation of the B-spline coefficients is made before the ice field drifts for the next time step. If the icebreaker position is not within a new cell, the ice field just drifts without updating the B-spline coefficients.

EXAMPLES

In order to show that B-splines can be used as a guidance model for an ice field, some examples are next presented. To make these examples, the MathWorks software MATLAB® and Simulink® is used. A one dimensional (1D) and a two dimensional (2D) example will be shown, considering the x-direction and y-direction, respectively. The 1D example of the ice field is included to provide insight on how the use of B-splines works, which is useful for understanding the 2D case.

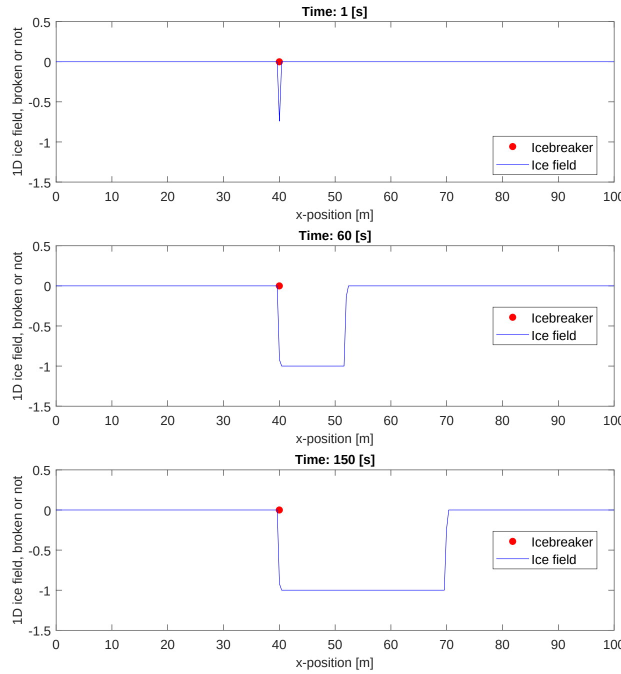

FIGURE 4: Three time instants for the 1D ice field, 1 [s], 60 [s], and 150 [s].

1D Example

For this example the x-axis is discretized into 0.2 [m] segments from 0 [m] to 100 [m]. For convenience the ice drift velocity is also set to 0.2 [m/s]. The B-splines are set as 3$^{\text{rd}}$ order with 2 points per interval, which will give a total of 501 knots over the whole area. The icebreaker in this example is stationary at position $x = 40$ [m], but it could easily have been set to move along the x-axis.

In Figure 4 the 1D ice field results are presented. It shows the results from three different time instants, 1 [s], 60 [s], and 150 [s], where we can see that the ice field changes its value to -1 when it passes through the icebreaker, indicating broken ice. This is exactly the intended behavior.

2D Example

The 2D ice field example is discretized into segments from 0 [m] to 1000 [m] in x-direction and -500 [m] to 500 [m] in y-direction, with a size of 15×15 [m^2] of the cells. This distance is chosen as it is approximately half the breadth of the icebreaker IB Oden [20]. The ice drift speed is the same as in the 1D example, 0.2 m/s, and the ice drift direction is from North to South. The

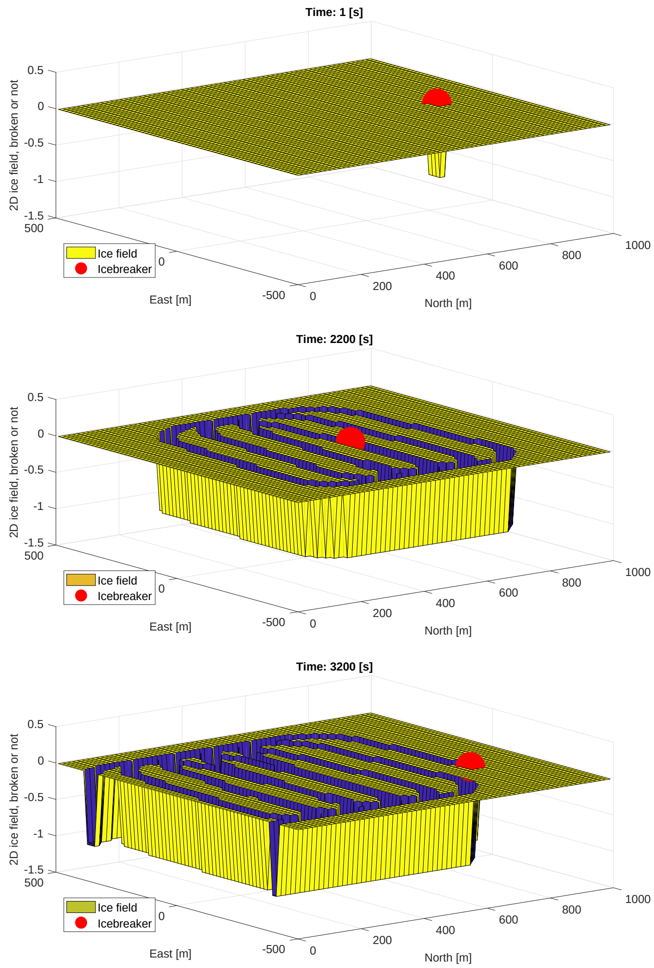

FIGURE 5: Three time instants for the 2D ice field, 1 [s], 2200 [s] and 3200 [s].

icebreaker is not stationary. Instead, it follows the racetrack path shown in Figure 3, with a radius of 400 meters to either side. The B-splines settings are the same as for the 1D example with 3^{rd} order with 2 points per interval, which will give a total of 4489 knots over the whole area.

To simulate the icebreaker, a predefined path is made with a given x- and y-position for each time step. This track makes the icebreaker keep a constant velocity throughout the track and simulation, and in this case the icebreaker velocity is set to 6 knots ($\approx$ 3.09 m/s). Figure 3 shows the racetrack-shaped icebreaker path used.

In Figure 5 the results for the 2D ice field are presented, showing the state at three time instants; 1 [s], 2200 [s], and 3200 [s]. From this figure it can be seen that the value in the cells change to a value of -1 when it has been visited by the icebreaker. Thereafter, this "broken ice" state value drifts to the

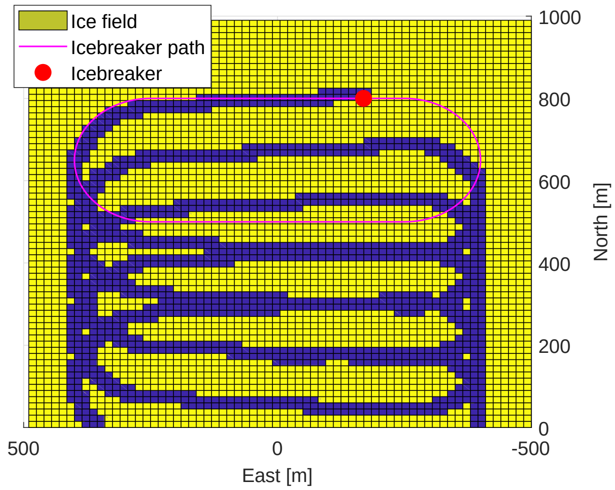

FIGURE 6: A cell view of the racetrack-shaped icebreaker path with a drifting surface, at time instant 3200 [s].

neighboring cells along the ice drift direction. Figure 6 shows the complete icebreaker track at 3200 seconds, where the dark path is the icebreaker footprint drifting with the ice and the solid line shows the actual icebreaker path. The whole icebreaker footprint, that drifts with the ice, can also be found in the last figure in Figure 5.

The computation time takes about 7 seconds each time the ice field is updated. These updates happen every time there is a change in one of the nodes, meaning nodes that have interacted with the icebreaker. Consequently, simulation over fine mesh can easily lead to extensive computational time. There is a need to look into this and see if the computational time can be lowered.

CONCLUSION

In this paper a suggestion on how to represent a simplified ice field in a guidance model, enabling path and maneuver planning for optimal IM operation, by the use of B-splines is presented. From the results of the 2D example, this ice field guidance model can be seen with both broken and intact ice, while still being a continuous surface. This means that the objective of representing the ice breaking state of the ice field has been fulfilled.

Further work is to reduce the computational time and see if there is a possibility of including long cracks that will be formed in between the IM track channels [2, 21, 22, 23, 24]. Then, this representation of a simplified ice field can be used in a path-planning algorithm to determine the optimal icebreaker path in a moving ice environment, in order to reduce the ice field into small enough ice floes and reduce the load on a protected structure. If this work is successful, this ice field guidance model can be included in a path-planning function and verified with an high-fidelity software, e.g., the Simulator for Arctic Marine Structures (SAMS) [9] for more extensive simulations.

ACKNOWLEDGMENT

This work was supported by the Research Council of Norway through the CRI SAMCoT, RCN prj. no. 203471, and partly through the CoE NTNU AMOS, RCN prj. no. 223254.

REFERENCES

[1] ISO/FDIS/19906, 2018. Petroleum and natural gas industries - Arctic offshore structures. International standard, International Organization for Standardization, Geneva, Switzerland.

[2] Hamilton, J. M., Holub, C., and Blunt, J., 2011. "Simulation of ice management fleet operations using two decades of beaufort sea ice drift and thickness time histories". *International Society of Offshore and Polar Engineers*.

[3] Hamilton, J., Holub, C., Blunt, J., Mitchell, D., and Kokkinis, T., 2011. "Ice management for support of arctic floating operations". In OTC Arctic Technology Conference, Offshore Technology Conference.

[4] Hamilton, J. M., 2011. "The Challenges of Deep-Water Arctic Development". *Int. Journal of Offshore and Polar Engineering,* ***21***(4), pp. 241–247.

[5] Lu, W., Heyn, H.-M., Lubbad, R., and Løset, S., 2018. "A large scale simulation of floe-ice fractures and validation against full-scale scenario". *International Journal of Naval Architecture and Ocean Engineering,* ***10***(3), may, pp. 393–402.

[6] Heyn, H.-M., Knoche, M., Zhang, Q., and Skjetne, R., 2017. "A system for automated vision-based sea-ice concentration detection and floe-size distribution indication from an icebreaker". In Volume 8: Polar and Arctic Sciences and Technology Petroleum Technology, American Society of Mechanical Engineers.

[7] Lubbad, R., Løset, S., Lu, W., Tsarau, A., and van den Berg, M., 2018. "An overview of the oden arctic technology research cruise 2015 (OATRC2015) and numerical simulations performed with SAMS driven by data collected during the cruise". *Cold Regions Science and Technology*, may.

[8] Lubbad, R., Løset, S., Lu, W., Tsarau, A., and van den Berg, M., 2018. "Simulator for arctic marine structures (SAMS)". In Volume 8: Polar and Arctic Sciences and Technology Petroleum Technology, ASME.

[9] Tsarau, A., van den Berg, M., Lu, W., Lubbad, R., and Løset, S., 2018. "Modelling results with a new simulator for arctic marine structures - SAMS". In Volume 8: Polar and Arctic Sciences and Technology; Petroleum Technology, American Society of Mechanical Engineers.

[10] Fossen, T. I., 2011. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd.

[11] Jørgensen, U., and Skjetne, R., 2015. "Online reconstruction of drifting underwater ice topography: The 2d case". *Asian Journal of Control,* ***17***(5), jan, pp. 1509–1521.

[12] Jørgensen, U., and Skjetne, R., 2013. "Real-time 3d reconstruction of underwater sea-ice topography by observations from a mobile robot in the arctic". *IFAC Proceedings Volumes,* ***46***(33), pp. 310–315.

[13] Schoenberg, I. J., 1946. "Contributions to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae". *Quarterly of Applied Mathematics,* ***4***(1), apr, pp. 45–99.

[14] Curry, H. B., and Schoenberg, I. J., 1947. "On spline distributions and their limits: the polya distributions". *Bulletin of the American Mathematical Society,* ***54***(1114).

[15] de Boor, C., 1972. "On calculating with B-splines". *Journal of Approximation Theory,* ***6***(1), jul, pp. 50–62.

[16] Brunet, F., 2010. "Contributions to Parametric Image Registration and 3D Surface Reconstruction". PhD thesis, Universite dAuvergne.

[17] de Boor, C., 2001. *A Practical Guide to Splines*. Springer-Verlag GmbH.

[18] Schumaker, L., 2015. *Spline Functions*. Cambridge University Press.

[19] de Boor, C., 1971. "Subroutine package for calculating with b-splines.".

[20] Kjerstad, Ø. K., Lu, W., Skjetne, R., and Løset, S., 2018. "A method for real-time estimation of full-scale global ice loads on floating structures". *Cold Regions Science and Technology*, mar.

[21] Lu, W., Lubbad, R., Løset, S., and Skjetne, R., 2016. "Parallel channel tests during ice management operations in the arctic ocean". In Arctic Technology Conference, Offshore Technology Conference.

[22] Lu, W., Lubbad, R., and Lset, S., 2015. "Tentative Fracture Mechanisms of the Parallel Channel Effect during Ice Management". In Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions.

[23] Lu, W., Lubbad, R., Shestov, A., and Løset, S., 2018. "Parallel channels' fracturing mechanism during ice management operations. part I: Theory". *Cold Regions Science and Technology,* ***156***, dec, pp. 102–116.

[24] Lu, W., Lubbad, R., and Løset, S., 2018. "Parallel channels fracturing mechanism during ice management operations. part II: Experiment". *Cold Regions Science and Technology,* ***156***, dec, pp. 117–133.

