An Ice-Drift Estimation Algorithm Using Radar and Ship Motion Measurements

Øivind K. Kjerstad, Sveinung Løset, Roger Skjetne, and Runa A. Skarbø

***Abstract*—This paper presents a novel automatic real-time remote sensing algorithm that uses radar images and global positioning satellite system measurements to estimate the ice-drift velocity vector in a region around a free-floating and potentially moving vessel. It is motivated by the low image frequency of satellite systems together with the inconvenience of deploying and retrieving ice trackers (beacons) on the ice. The algorithm combines radar image processing with two Kalman filters to produce the estimated local drift vector decoupled from the ship motion. The proposed design is verified using a full-scale data set from an ice management operation north of Svalbard in 2015. It is found that the performance of the algorithm is comparable with that of trackers on the ice.**

***Index Terms*—Ice, Remote Sensing, Radar, Marine vehicles, Arctic, Kalman filtering.**

I. Introduction

Sea-ice occurs in about 10% of the world ocean's surface, and lies mostly in the Arctic and Antarctic Seas. It grows and melts under influence of solar, atmospheric, oceanic, and tidal forcing where the ice covers break up, open, and close as drifting ice floe fields [1]. Operating fixed or floating structures in such environments impose several challenges related to the movement of the sea-ice. Therefore, regardless of the purpose of the human presence, it is important to monitor the sea-ice and plan according to its movement to control the operational risk level.

On a large scale, spaceborne satellite systems with optical and microwave imagery, and synthetic aperture radars (SARs) provide excellent images that are valuable in planning and carrying out operations. From these, information about ice concentration, ice cover composition, and ice-drift on a large scale can be obtained [2]. However, the downside of these intelligence sources is their data frequency. This may be on a few-times-a-day scale, and the real-time availability of the data in the range of several hours [3]. Thus, satellites in low Polar orbit have limited capability to monitor the ice-drift for online operational decision support, as the ice-drift is prone to significant change within a few hours [4, 5]. To counter this, real-time ice-drift monitoring is typically implemented by either physically deploying (and retrieving) reference GNSS sensors on large ice floes, or by an ice observer crew member evaluating the ice cover and ship status to provide a coarse description of the local situation. Both of these methods have notable limitations. Firstly, it may be difficult for a human on a moving platform to perceive the slow movement of the ice with good accuracy [6], not to mention difficulties with darkness, harsh weather, and daily routines. Secondly, placing and retrieving sensor and communication systems with batteries on the ice will in many cases require helicopter operations, often not possible to conduct, and in other cases it may simply not be allowed from an environmental perspective. This motivates the development of a robust onboard remote sensing system capable of tracking the ice-drift velocity vector in real-time during day and night, and in good and harsh weather.

To improve local ice-drift monitoring we propose an automatic real-time algorithm that estimates the ice-drift vector in the vicinity of the vessel with high temporal resolution. Here, local refers to an area with a few (0.5 to 6) nautical miles in radius. The algorithm uses image processing techniques to automatically detect and track the motion of N distinctive features (DF) in a north fixed radar plan position indicator (PPI) image, and two Kalman filters to select DFs and decouple the vessel motion. Essentially, the proposed algorithm is a target tracking system that combines multiple targets to provide an estimate of the ice-drift in the area covered by the radar. Target tracking of multiple objects using marine navigational radar is not new, and several systems exists (see e.g. [7]). However, to our best knowledge, no algorithm provides explicit ice-drift measures and predictions decoupled from a moving platform like a ship. From land-based or freely drifting platforms some similar works exist [8].

The proposed algorithm is investigated using a dataset from the Oden Arctic Technology Research Cruise in 2015 (OATRC15), where two Swedish icebreakers, the Oden and the Frej, conducted ice management (IM) trials in the Arctic sea-ice north of Svalbard. On this expedition we had several beacons deployed on the ice.

Mathematical notation: Bold font face is used to denote vectors and matrices, and normal face is used for scalars. The subscript k describes the discrete time instance of a variable. $\mathbf{A} \circ \mathbf{B}$ denotes the Schur product, or entrywise product, between two matrices $\mathbf{A}$ and $\mathbf{B}$ of equal size. In the Kalman filter the notation $\mathrm{k}|\mathrm{k}-1$ denotes the *a priori*, or predicted, estimate, and $\mathrm{k}|\mathrm{k}$ denotes the *a posteriori*, or updated estimate.

II. Modeling and

Ø. K. Kjerstad is with the Department of Arctic Technology, The University Center in Svalbard, Norway, e-mail: oivind.kjerstad@unis.no.

S. Løset, R. Skjetne, and R. A. Skarbø are with the Norwegian University of Science and Technology, Trondheim, Norway.

Manuscript sent December 14, 2016

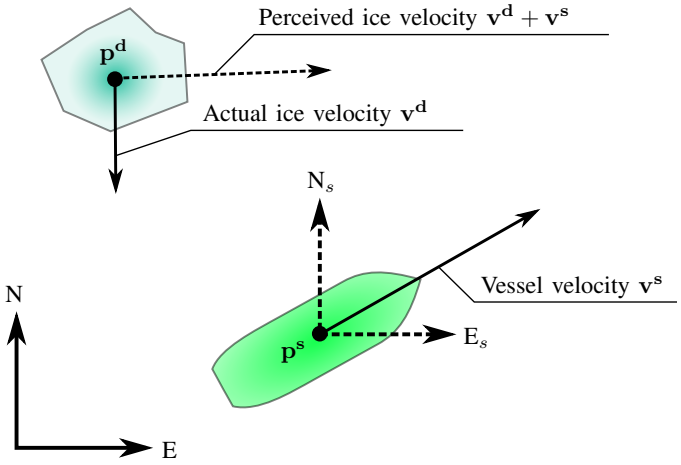

Fig. 1. The ship and DF kinematics shown in the North-East-Down frame.

PROBLEM FORMULATION

This paper considers the kinematic scenario illustrated in Figure 1, where the vessel position is measured in a geo-fixed North-East-Down (NED) frame (by global navigation satellite system (GNSS) e.g. the global positioning system (GPS)), and the DFs are captured in a vessel-centered NED frame image by the onboard radar (an example is given in Figure 2). The key problem of determining the ice-drift velocity from the latter is that the observed motion of the DFs are a superposition of the vessel and DF motion (due to the fact that the radar antenna is shipborne). Hence, the objective is to estimate and predict the geo-fixed NED ice-drift velocity vector $\mathbf{v}^{\text{ice}} \in \mathbb{R}^2$ in the radar-observed area.

Since radars do not distinguish between different objects, any echo is a potential DF. By assuming that most echoes originate from drifting sea-ice, we propose the algorithm structure shown in Figure 3:

1) Find and track the relative position of N independent DFs in the radar image.
2) Estimate the geo-fixed velocity of the N DFs individually.
3) Separate non-ice DFs from ice DFs.
4) Estimate $\mathbf{v}^{\text{ice}}$ using the ice DFs.

To express the vessel motion we apply the following two degrees of freedom (DOF) NED frame model,

$$\mathbf{p}^{\text{s}}_{\text{k+1}} = \mathbf{p}^{\text{s}}_{\text{k}} + \text{h}\mathbf{v}^{\text{s}}_{\text{k}} + \mathbf{w}^{\text{sp}}_{\text{k}} \tag{1a}$$
$$\mathbf{v}^{\text{s}}_{\text{k+1}} = \mathbf{v}^{\text{s}}_{\text{k}} + \mathbf{w}^{\text{sv}}_{\text{k}} \tag{1b}$$
$$\mathbf{y}^{\text{s}}_{\text{k}} = \mathbf{p}^{\text{s}}_{\text{k}} + \mathbf{n}^{\text{s}}_{\text{k}}, \tag{1c}$$

where $\mathbf{p}^{\text{s}} \in \mathbb{R}^2$ is the vessel position, $h \in \mathbb{R}$ is the time-step of the model, $\mathbf{v}^{\text{s}} \in \mathbb{R}^2$ is the vessel velocity, $\mathbf{w}^{\text{sp}} \in \mathbb{R}^2$ and $\mathbf{w}^{\text{sv}} \in \mathbb{R}^2$ are process noise vectors, $\mathbf{y}^{\text{s}} \in \mathbb{R}^2$ is the output measurement, and $\mathbf{n}^{\text{s}} \in \mathbb{R}^2$ is measurement noise. A DF is modeled similarly,

$$\mathbf{p}^{\text{d}}_{\text{k+1}} = \mathbf{p}^{\text{d}}_{\text{k}} + \text{h}\mathbf{v}^{\text{d}}_{\text{k}} + \mathbf{w}^{\text{dp}}_{\text{k}} \tag{2a}$$
$$\mathbf{v}^{\text{d}}_{\text{k+1}} = \mathbf{v}^{\text{d}}_{\text{k}} + \mathbf{w}^{\text{dv}}_{\text{k}} \tag{2b}$$
$$\mathbf{y}^{\text{d}}_{\text{k}} = \mathbf{p}^{\text{d}}_{\text{k}} - \mathbf{p}^{\text{s}}_{\text{k}} + \mathbf{n}^{\text{d}}_{\text{k}}, \tag{2c}$$

where $\mathbf{p}^{\text{d}} \in \mathbb{R}^2$ is the DF position, $\mathbf{v}^{\text{d}} \in \mathbb{R}^2$ is its velocity, $\mathbf{w}^{\text{dp}} \in \mathbb{R}^2$ and $\mathbf{w}^{\text{dv}} \in \mathbb{R}^2$ are process noise vectors, $\mathbf{y}^{\text{d}} \in \mathbb{R}^2$ is the model output measurement, and $\mathbf{n}^{\text{d}} \in \mathbb{R}^2$ is measurement noise. Notice that in (2c) the term $\mathbf{p}^{\text{s}}_{\text{k}}$ originates from measuring the DF from the vessel.

To simplify the notation we write the complete ship and N DF model in state space form as

$$\mathbf{x}_{\text{k+1}} = \mathbf{F}\mathbf{x}_{\text{k}} + \mathbf{E}\mathbf{w}_{\text{k}} \tag{3a}$$
$$\mathbf{y}_{\text{k}} = \mathbf{H}\mathbf{x}_{\text{k}} + \mathbf{D}\boldsymbol{\xi}_{\text{k}} \tag{3b}$$

where $\mathbf{x} \in \mathbb{R}^{4\text{N+4}}$ is a vector of the N DF states and the vessel states. This is defined as

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}^{\text{d1}} & \mathbf{v}^{\text{d1}} & \dots & \mathbf{p}^{\text{dN}} & \mathbf{v}^{\text{dN}} & \mathbf{p}^{\text{s}} & \mathbf{v}^{\text{s}} \end{bmatrix}^\top, \tag{4a}$$

where the superscript numbering refer to a particular tracked DF. Further in (3), $\mathbf{w} \in \mathbb{R}^{4\text{N+4}}$ is the overall process noise vector, and $\boldsymbol{\xi} \in \mathbb{R}^{4\text{N+4}}$ is the measurement noise vector. These are defined as

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^{\text{dp1}} & \mathbf{w}^{\text{dv1}} \dots \mathbf{w}^{\text{dpN}} & \mathbf{w}^{\text{dvN}} & \mathbf{w}^{\text{sp}} & \mathbf{w}^{\text{sv}} \end{bmatrix}^\top, \tag{4b}$$
$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{n}^{\text{d1}} & \mathbf{n}^{\text{d2}} & \dots & \mathbf{n}^{\text{dN}} & \mathbf{n}^{\text{s}} \end{bmatrix}^\top. \tag{4c}$$

As in (4a), superscript numbering refer to a particular tracked DF. The state-space matrices $\mathbf{E} \in \mathbb{R}^{(4\text{N+4}) \times (4\text{N+4})}$ and $\mathbf{D} \in \mathbb{R}^{(4\text{N+4}) \times (4\text{N+4})}$ are identity matrices, and $\mathbf{F} \in \mathbb{R}^{(4\text{N+4}) \times (4\text{N+4})}$ and $\mathbf{H} \in \mathbb{R}^{(2\text{N+2}) \times (4\text{N+4})}$ are defined as

$$\mathbf{F} = \begin{bmatrix} \mathbf{I} & h\mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & & & \ddots & & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & h\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{5a}$$

$$\mathbf{H} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & & \ddots & & & \vdots \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}. \tag{5b}$$

Based on the natural motion of the sea-ice [1, 4, 5], we propose the following simplified kinematic ice-drift velocity model,

$$\mathbf{v}^{\text{ice}}_{\text{k+1}} = \mathbf{a}_{\text{k}} \sin(\boldsymbol{\omega}_{\text{k}} t + \boldsymbol{\phi}_{\text{k}}) + \mathbf{b}_{\text{k}} + h\mathbf{w}^{\text{ice}}_{\text{k}}, \tag{6}$$

where $t \in \mathbb{R}$ denotes time, and $\mathbf{a}_{\text{k}}$, $\boldsymbol{\omega}_{\text{k}}$, $\boldsymbol{\phi}_{\text{k}}$, $\mathbf{b}_{\text{k}}$, and $\mathbf{w}^{\text{ice}} \in \mathbb{R}^2$ represent the amplitude, frequency, phase, signal offset, and process noise, respectively. The first four are assumed to be unknown and slowly varying. Thus, they are modeled as

$$\mathbf{a}_{\text{k+1}} = \mathbf{a}_{\text{k}} + h\mathbf{w}^{\text{a}}_{\text{k}} \tag{7a}$$
$$\boldsymbol{\omega}_{\text{k+1}} = \boldsymbol{\omega}_{\text{k}} + h\mathbf{w}^{\omega}_{\text{k}} \tag{7b}$$
$$\boldsymbol{\phi}_{\text{k+1}} = \boldsymbol{\phi}_{\text{k}} + h\mathbf{w}^{\phi}_{\text{k}} \tag{7c}$$
$$\mathbf{b}_{\text{k+1}} = \mathbf{b}_{\text{k}} + h\mathbf{w}^{\text{b}}_{\text{k}}, \tag{7d}$$

where $\mathbf{w}^{\text{a}}_{\text{k}}$, $\mathbf{w}^{\omega}_{\text{k}}$, $\mathbf{w}^{\phi}_{\text{k}}$, and $\mathbf{w}^{\text{b}}_{\text{k}} \in \mathbb{R}^2$ are process noise terms. On compact form (6)-(7) can be written

$$\mathbf{z}_{\text{k+1}} = \mathbf{G}(\mathbf{z}_{\text{k}}, h) + h\mathbf{w}^{\text{z}}_{\text{k}} \tag{8a}$$
$$\mathbf{y}^{\text{ice}}_{\text{k}} = \mathbf{C}_{\text{k}}\mathbf{x}_{\text{k}}, \tag{8b}$$

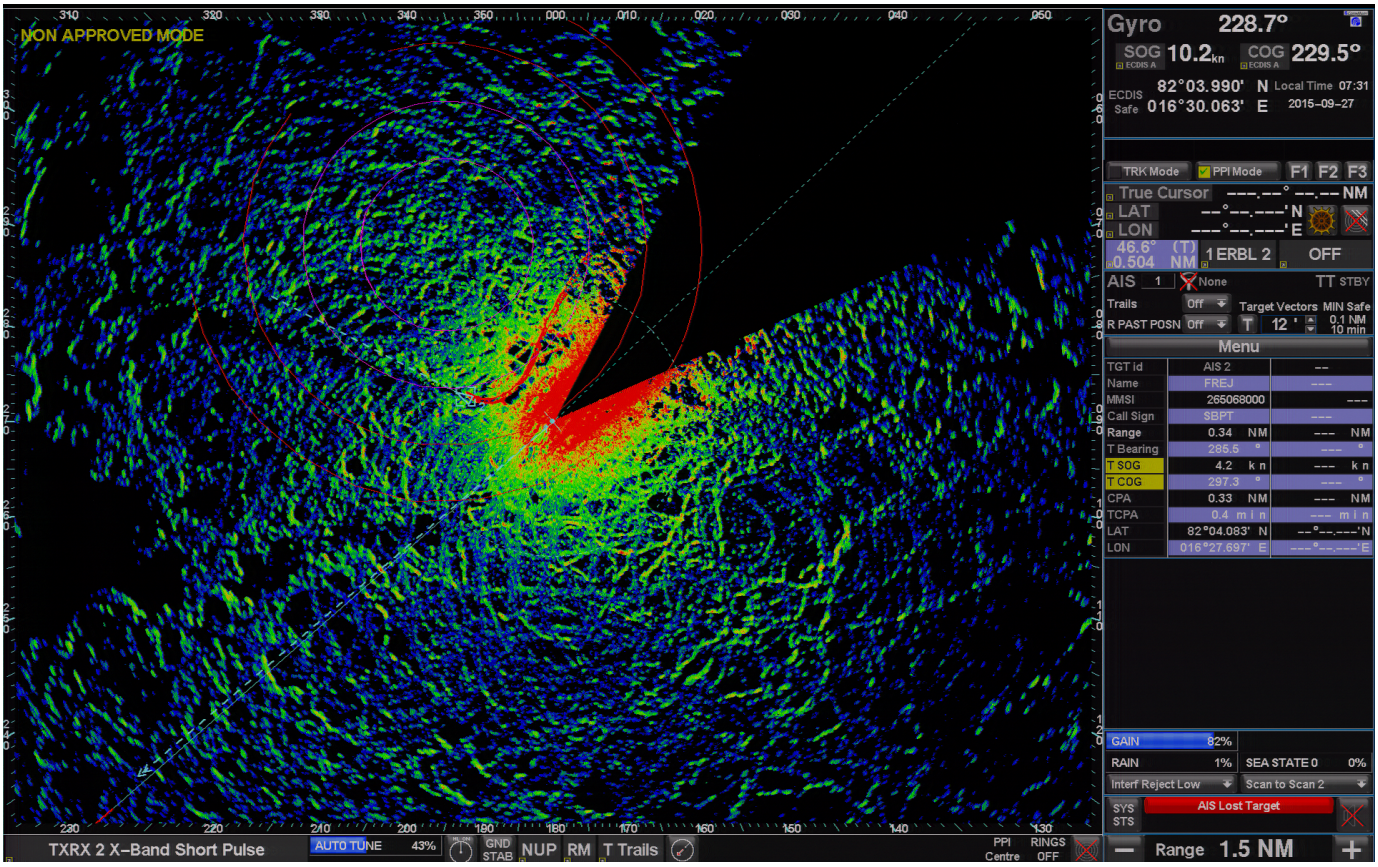

Fig. 2. A unprocessed radar image captured from the operator workstation during ice management trials at the OATRC15.

where $\mathbf{z}$ and $\mathbf{w}^{\text{ice}}$ are defined as

$$\mathbf{z} = \begin{bmatrix} \mathbf{v}^{\text{ice}} & \mathbf{a} & \boldsymbol{\omega} & \boldsymbol{\phi} & \mathbf{b} \end{bmatrix}^{\top}, \tag{8c}$$

$$\mathbf{w}^{\text{ice}} = \begin{bmatrix} \mathbf{w}^{\text{z}} & \mathbf{w}^{\text{a}} & \mathbf{w}^{\omega} & \mathbf{w}^{\phi} & \mathbf{w}^{\text{b}} \end{bmatrix}^{\top}. \tag{8d}$$

In (8b), $\mathbf{C}_{\text{k}} \in \mathbb{R}^{\Upsilon \times (4N+4)}$ is a selection matrix calculated based on the number of DFs complying with a criterion for inclusion in calculation of $\mathbf{z}$ (from which the variable $\Upsilon \in \mathbb{R}$ follows). The definition of this will be presented later.

The remainder of this paper is devoted to presenting and experimentally testing the proposed algorithm. Section III presents the algorithm to estimate $\mathbf{x}_{\text{k}}$ and $\mathbf{z}_{\text{k}}$ based on radar images and position measurements. Section IV investigates the performance of the proposed estimation system using the mentioned OATRC15 dataset. Finally, Section V concludes the study.

III. Algorithm Design

A. Image processing

As seen in Figure 3, the image processing of the radar images acts as a ship-relative position sensor for the N DFs. Together with vessel position measurements, it enables determination of $\mathbf{y}_{\text{k}}$ in (3), which is needed for the estimation of $\mathbf{x}_{\text{k}}$ and $\mathbf{z}_{\text{k}}$.

The image processing considered in this paper is built on the well-known and proven concept of corner detection, where we define a DF as a point for which there are two dominant and different edge directions in a local neighborhood. This implies that a DF may be a corner, but it can also be an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal [9]. Although corner detection methods may provide the DF locations in the radar images, they do not provide a robust manner of linking the DFs found in one image to those found in the next. Hence, it does not provide consistent tracking between radar frames. Achieving this is inspired by optical flow methods such as [10], but instead of computing the DF velocities directly we leave this to the state estimation sub-algorithm and focus on obtaining a robust DF match. This will avoid direct differentiation, which is highly susceptible to noise.

The three modules of the *Image Processing* block of Figure 3 are described and explained below.

1) Image preparation: As mentioned above, the data applied in this paper are derived from the OATRC15, and the radar images are in the format seen in Figure 2. To enable detection and tracking of DFs on such images, pre-processing to a suitable format is needed. This is done in the *Prepare Image* module seen in Figure 3. Algorithm 1 provides pseudo-code to explain the module.

Following Algorithm 1 the original radar image $\mathbf{F}_{\text{r}} \in \mathbb{R}^{1920\times\ 1200}$ is cropped to remove all non-PPI graphics, ob-

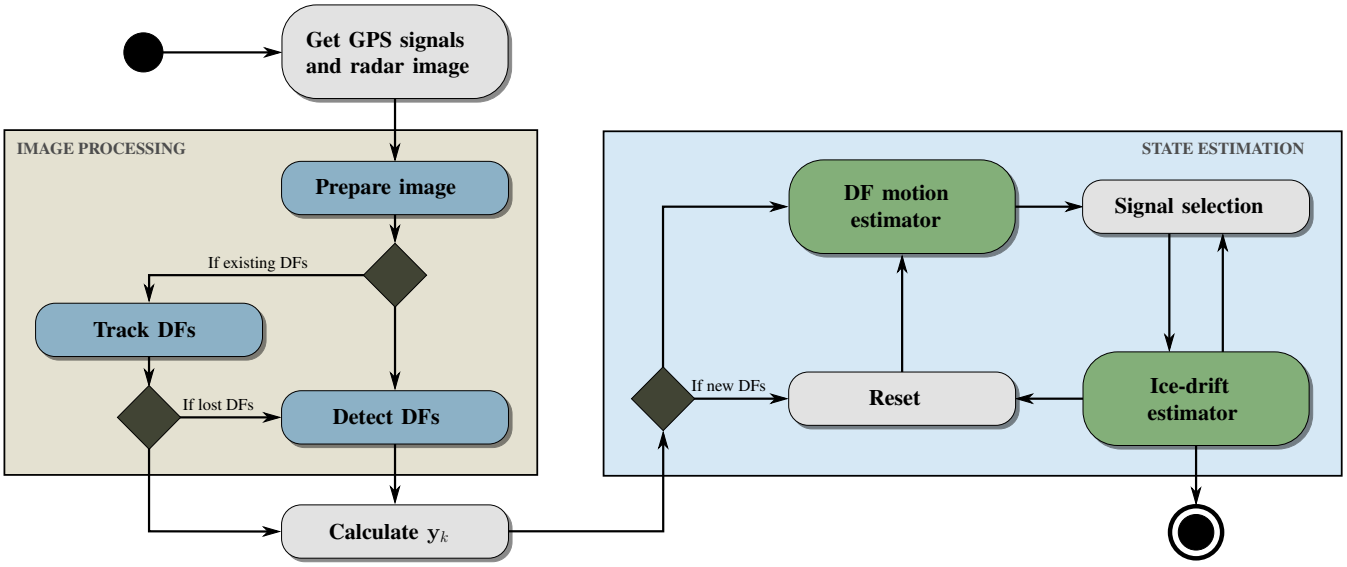

Fig. 3. An overview of one iteration of the ice-drift velocity vector estimation algorithm.

Algorithm 1 Pseudo-code for preparing the radar frame for further processing.

function Prepare Image

Input: RGB radar image, $\mathbf{F}_\mathrm{r}$
Output: Cornermetic matrix, $\mathbf{\Omega}$

1: *Get* $\mathbf{F}_c$ *by cropping* $\mathbf{F}_r$ *to PPI*
2: *Compute* $\mathbf{F}_w$ *from* $\mathbf{F}_c$
3: *Filter* $\mathbf{F}_w$ *and compute* $\mathbf{\Omega}$

end function

taining $\mathbf{F}_\mathrm{c} \in \mathbb{R}^{q_\mathrm{x} \times q_\mathrm{y}}$, where q_x, $q_\mathrm{y} \in \mathbb{R}$ are the cropped image pixel dimensions. Secondly, $\mathbf{F}_\mathrm{c}$ is converted to gray-scale, $\mathbf{F}_\mathrm{w} \in \mathbb{R}^{q_\mathrm{x} \times q_\mathrm{y}}$, in order to facilitate DF detection by a corner method. Thirdly, we scale out the image center by

$$\mathbf{F}_\mathrm{w} = \mathbf{F}_\mathrm{c} \circ \mathbf{Z} \tag{9}$$

where $\mathbf{Z} \in \mathbb{R}^{q_\mathrm{x} \times q_\mathrm{y}}$ is a weighting matrix computed by

$$\mathbf{Z} = \mathbf{J}_1 - \frac{1}{max(\mathbf{\Gamma})}\mathbf{\Gamma} \tag{10a}$$

$$\mathbf{\Gamma}(i,j) = \frac{1}{\sqrt{2\pi}} e^{-(i-i_0)^{\theta_x} - (j-j_0)^{\theta_y}}, \tag{10b}$$

where $\mathbf{J}_1 \in \mathbb{R}^{q_\mathrm{x} \times q_\mathrm{y}}$ is unit matrix, $\mathbf{\Gamma} \in \mathbb{R}^{q_\mathrm{x} \times q_\mathrm{y}}$ is a weighting matrix, $i \in \mathbb{R}_{\geq 0}$ and $j \in \mathbb{R}_{\geq 0}$ are pixel rows and columns, respectively, i_0 and j_0 denote the image center, and $\theta_x \in \mathbb{R}_{>0}$ and $\theta_y \in \mathbb{R}_{>0}$ are design weights. The foremost reason for weighting down the center close to the ship, is the reduced radar data quality in this area. Figure 2 illustrates this well.

Before computing the cornermetric matrix $\mathbf{\Omega} \in \mathbb{R}^{q_\mathrm{x} \times q_\mathrm{y}}$, we filter $\mathbf{F}_\mathrm{r}$ with a median window filter to remove noise while preserving edges. Finally, $\mathbf{\Omega}$ is computed by the method described in [11]. Figure 4 shows the Prepare Image output for the radar image of Figure 2. Notice that it contains DFs that are not originating from the ice, but from overlay graphics. Such will be mitigated and discussed further in Section III-C.

2) Detecting DFs: The *Detect DFs* module of Figure 3 takes $\mathbf{\Omega}$ and pinpoints coordinates for $n \in \{1, \ldots, N\}$ DFs with sub-pixel precision. It is used at initialization and whenever one or more DF are lost (e.g. not re-found in the next frame by the *Track DFs* module to be discussed below). The Detect DFs module will search for and insert new DFs to keep the number of tracked DFs constant at N. Algorithm 2 provides pseudo-code to explain the module.

Algorithm 2 Pseudo-code for detecting DFs using the cornermetric matrix $\mathbf{\Omega}$.

function Detect DFs

Input: Cornermetric matrix, $\mathbf{\Omega}$
Number of DFs to detect, n
List of existing DFs, $\mathbf{L}_\mathrm{e}$
Output: List of existing DFs, $\mathbf{L}_\mathrm{e}$

1: **for** $1 : n$ **do**
2: **while** DF not found **do**
3: *Find DF candidate in* $\mathbf{\Omega}$
4: **if** DF $\notin \mathbf{L}_\mathrm{e}$ **then**
5: *Compute sub-matrix* $\mathbf{W}_w$
6: *Compute centroid point* (C_x, C_y)
7: *Store* (C_x, C_y) *in* $\mathbf{L}_\mathrm{e}$
8: **end if**
9: **end while**
10: **end for**

end function

Following Algorithm 2, for each of the n DFs to be detected, the Detect DFs module will search for a DF candidate in $\mathbf{\Omega}$. This is done by iteratively selecting the highest non-evaluated peak $\gamma \in \mathbb{R}$

$$\gamma = max(\mathbf{\Omega}(i,j)) \tag{11}$$

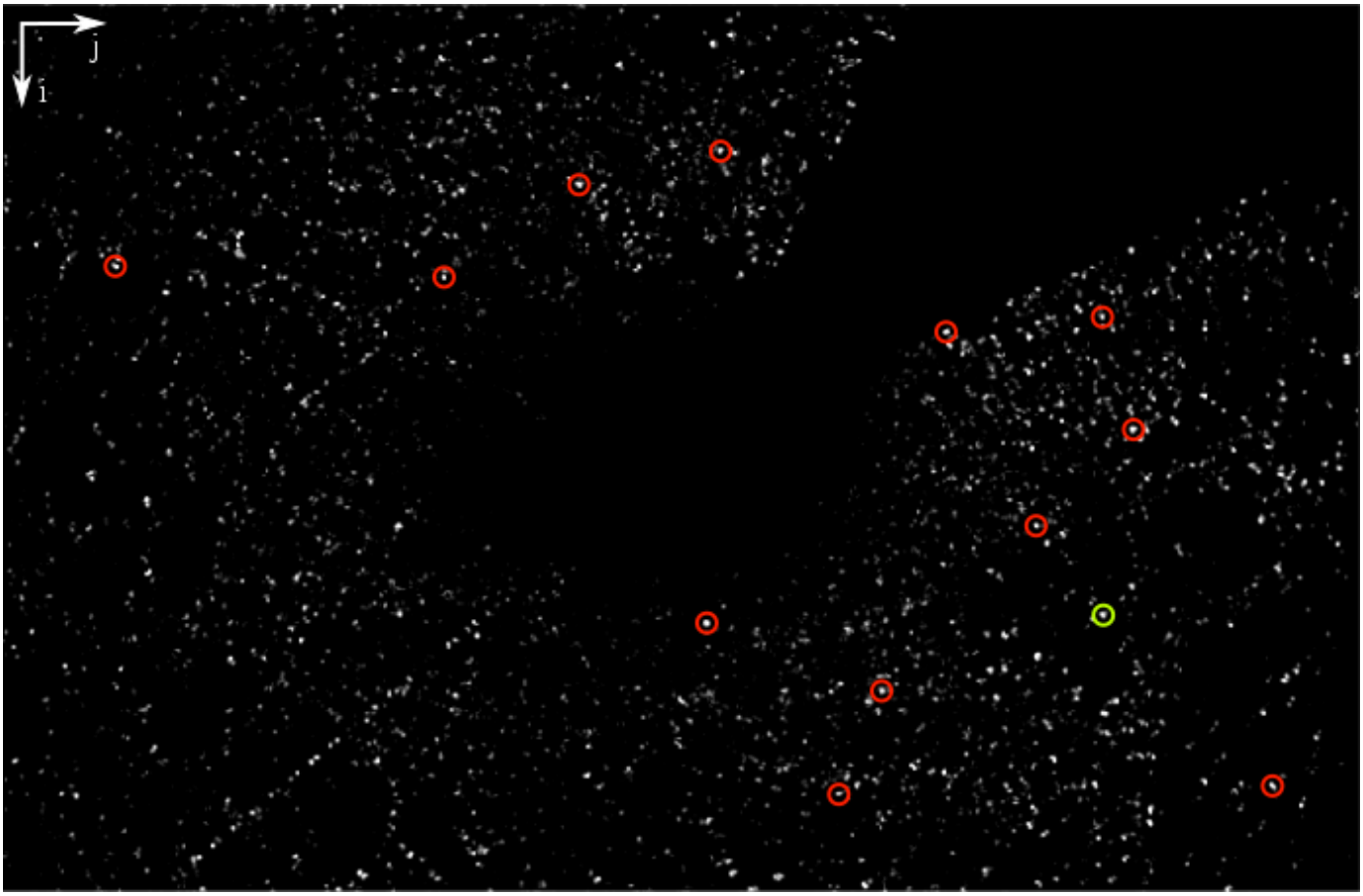

Fig. 4. Visualization of the *Prepare Image* module cornermetric matrix output together with several DFs found by the *Detect DFs* module for the radar image shown in Figure 2. The white spots show potential DFs, and the circles mark selected DFs. The brightness and contrast have been adjusted.

and enforcing the following criteria on the selected peak:

- Minimum distance to the matrix edges.
- Minimum distance from all DFs in $\mathbf{L}_\mathrm{e}$.

Although all γ values are associated with integer coordinates (i, j), they are typically a part of a larger peak footprint area in $\mathbf{\Omega}$. Here we define a DF as one such area, and its sub-pixel coordinates is the centroid of that area. This is computed by

$$C_\mathrm{x} = \frac{\sum_{i=1}^{q_\mathrm{w}} \sum_{j=1}^{q_\mathrm{w}} \mathbf{W}_\mathrm{w}(i,j) i}{\sum_{i=1}^{q_\mathrm{w}} \sum_{j=1}^{q_\mathrm{w}} \mathbf{W}_\mathrm{w}(i,j)} + O_\mathrm{x}(\gamma) \tag{12a}$$

$$C_\mathrm{y} = \frac{\sum_{i=1}^{q_\mathrm{w}} \sum_{j=1}^{q_\mathrm{w}} \mathbf{W}_\mathrm{w}(i,j) j}{\sum_{i=1}^{q_\mathrm{w}} \sum_{j=1}^{q_\mathrm{w}} \mathbf{W}_\mathrm{w}(i,j)} + O_\mathrm{y}(\gamma), \tag{12b}$$

where $(C_\mathrm{x}, C_\mathrm{y})$ is the coordinates in $\mathbf{\Omega}$, $\mathbf{W}_\mathrm{w} \in \mathbb{R}^{q_\mathrm{w} \times q_\mathrm{w}}$ is a modified sub-matrix of $\mathbf{\Omega}$ centered on γ, $q_\mathrm{w} \in \mathbb{R}$ is the sub-matrix size, and $O_\mathrm{x}(\gamma)$ and $O_\mathrm{y}(\gamma)$ give the offset of $\mathbf{W}_\mathrm{w} \in \mathbb{R}^{q_\mathrm{w} \times q_\mathrm{w}}$ in $\mathbf{\Omega}$. $\mathbf{W}_\mathrm{w}$ is computed as

$$\mathbf{W}_\mathrm{w} = \mathbf{W} \circ \mathbf{W}_\mathrm{b}, \tag{13a}$$

where $\mathbf{W} \in \mathbb{R}^{q_\mathrm{w} \times q_\mathrm{w}}$ is the unmodified sub-matrix of $\mathbf{\Omega}$ centered on γ, and $\mathbf{W}_b \in \mathbb{R}^{q_\mathrm{w} \times q_\mathrm{w}}$ is a mask matrix given by

$$\mathbf{W}_\mathrm{b}(i,j) = \begin{cases} 1 \text{ if } & \mathbf{W}(i,j) > \kappa_b \gamma \\ 0 \text{ else,} & \end{cases} \tag{13b}$$

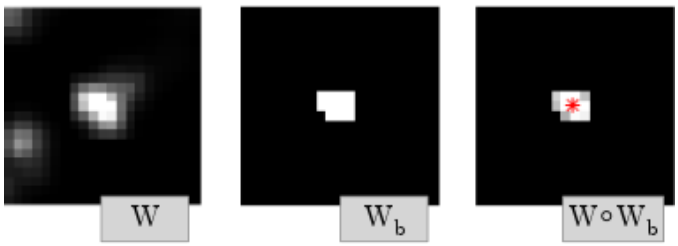

Fig. 5. The detection results of the DF in the green circle seen in Figure 4. The red star in the right image shows the computed weighted centroid.

where $0 < \kappa_b < 1$ is a static threshold design value.

If there are more than one unconnected area in $\mathbf{W}_\mathrm{b}$, the area encompassing γ is preserved and the pixel values of the others are set to 0. Figure 5 shows the different stages of finding the DF coordinates for the DF with the green circle in Figure 4. This sub-entry precision, using the whole DF footprint in $\mathbf{\Omega}$, is beneficial for precise state estimation, as it removes the discrete nature of pixels while increasing robustness to noise of the point measurement.

3) Tracking DFs: The *Track DFs* module of Figure 3 updates the coordinates of the DFs in $\mathbf{L}_e$ and determines if DFs are lost and new needs to be detected. It is applied at each iteration

besides initialization. Algorithm 3 provides pseudo-code to explain the module.

Algorithm 3 Pseudo-code for tracking DFs using the cornermetric matrix $\mathbf{\Omega}$.

function Track DFs

Input: Cornermetric matrix, $\mathbf{\Omega}$
List of existing DFs, $\mathbf{L}_\text{e}$
Output: List of existing DFs, $\mathbf{L}_\text{e}$
Number of DFs to detect, n

```
n = 0
for DFs in L_e do
    Compute DF projection
    if not lost then
        Compute sub-matrix W_w
        Compute centroid point (C_x, C_y)
        Store (C_x, C_y) in L_e
    else
        n=n+1
    end if
end for
```

end function

Tracking the DFs is achieved by solving an image registration problem to determine the translation of a sub-matrix $\mathbf{\Lambda} \in \mathbb{R}^{q_\Lambda \times q_\Lambda}$ in $\mathbf{\Omega}$ for each of the N DFs from one radar image frame to the next. To do this we apply the registration cross-correlation method described in [12]. This obtains an initial estimate of the cross-correlation peak by the fast Fourier transform (FFT) and refines the shift estimation by up-sampling the discrete Fourier transform (DFT) in a small neighborhood of that estimate by means of a matrix-multiply DFT. By this, it allows all the peaks in $\mathbf{\Lambda}$ to be used when computing the cross-correlation. Then, it is evaluated if the DF is considered found or lost. A DF may be lost in two ways:

- If its position is too close to the $\mathbf{\Omega}$ edge.
- If the peak value γ is lower than a threshold $\gamma_0 = \kappa_\gamma \sigma_\Omega$, where σ_Ω is the mean value of $\mathbf{\Omega}$ and $\kappa_\gamma > 0$ is a design threshold factor.

Following Algorithm 3, when a DF does not comply with either of the two criteria above it is considered persistent, and the procedure described in Section III-A2 to determine the exact position is applied. This step is needed to avoid numerical drift of the position values as the image registration algorithm will not give an absolute position, but a translation of the sub-matrix $\mathbf{\Lambda}$. If the DF is lost, it is registered by an increment increase in n, which if larger than zero triggers the Detect DFs module to find n new DFs (as seen in Figure 3).

B. Conversions of measurement

Since the output of the image processing is $\mathbf{L}_\text{e}$ is given in pixels and the vessel GPS position is in latitude and longitude, a conversion is needed to comply with the NED frame definition of (3) and (8). With respect to Figure 3, this is done in the block in the lower left corner.

To determine the vessel's NED coordinates from the GPS longitude and latitude measurement a coordinate transformation with respect to a set NED origin is needed. Such can be found in e.g. [13]. The origin of the NED frame (in longitude and latitude values) are typically at system initialization, or selected by an operator to best suit the operation.

To comply with (2c), the individual DF measurement vectors, $\mathbf{y}_k^d := \begin{bmatrix} y_x^d & y_y^d \end{bmatrix}^\top$, should be vessel relative in the NED frame. Thus, it is calculated from the pixel values of $\mathbf{L}_\text{e}$ at each time-step as

$$y_x^d = -C_R R_r (L_{ex} - x_\Omega) \tag{14a}$$
$$y_y^d = C_R R_r (L_{ey} - y_\Omega), \tag{14b}$$

where $C_R \in \mathbb{R}$ is a conversion coefficient, $R_r \in \mathbb{R}$ is the radar range set by the operator, $L_{ex}, L_{ey} \in \mathbb{R}$ are the ship-relative pixel position values of the DF (given in $\mathbf{L}_e$), and $x_\Omega, y_\Omega \in \mathbb{R}$ are the vessel pixel position in $\mathbf{\Omega}$. The sign of (14a) is due to the rotation of the row-column coordinate system of $\mathbf{\Omega}$ to NED (see Figure 4).

C. State estimation

A state estimator is a filter structure that is capable of reconstructing the complete system state online, using output measurement and system models (given that the system is observable [14]). In this paper we consider such because they allow model-based estimation, avoiding direct differentiation of the position signals from the image processing and ship position measurements. Another beneficiary property of state estimators is their ability to perform *dead reckoning*. That is, to produce reliable estimates for some time (based on the mathematical model alone) without measurement updates correcting the model.

We estimate $\mathbf{x}_\text{k}$ and $\mathbf{z}_\text{k}$ based on (3) and (8) using two Kalman filters (KF) in cascade, connected by handling logic for resetting DF estimates. With respect to Figure 3, these are the *DF motion estimator* and the *ice-drift estimator*. The former estimates the position and velocity of the vessel and individual DFs, and the latter fuses a subset of the DF velocity signals to form the ice-drift vector. Here the algorithms are presented for completeness without in-depth details. For further details see [14].

1) The DF motion estimator: As mentioned above, (3) is applied to estimate the position and motion of the DFs and the vessel. Since (3) is linear and time invariant (LTI), a linear

KF is applied to estimate $\mathbf{x}_\mathrm{k}$. This is written as

$$\hat{\mathbf{x}}_{\mathrm{k|k\text{-}1}} = \mathbf{F}\hat{\mathbf{x}}_{\mathrm{k\text{-}1|k\text{-}1}} \tag{15a}$$
$$\mathbf{P}_{\mathrm{k|k\text{-}1}} = \mathbf{F}\mathbf{P}_{\mathrm{k\text{-}1|k\text{-}1}}\mathbf{F}^\top + \mathbf{Q}_\mathrm{k} \tag{15b}$$
$$\tilde{\mathbf{y}}_\mathrm{k} = \mathbf{y}_\mathrm{k} - \mathbf{H}\hat{\mathbf{x}}_{\mathrm{k|k+1}} \tag{15c}$$
$$\mathbf{K}_\mathrm{k} = \mathbf{P}_{\mathrm{k|k\text{-}1}}\mathbf{H}^\top\big(\mathbf{H}\mathbf{P}_{\mathrm{k|k\text{-}1}}\mathbf{H}^\top + \mathbf{R}_\mathrm{k}\big)^{-1} \tag{15d}$$
$$\hat{\mathbf{x}}_{\mathrm{k|k}} = \hat{\mathbf{x}}_{\mathrm{k|k\text{-}1}} + \mathbf{K}_\mathrm{k}\tilde{\mathbf{y}}_\mathrm{k} \tag{15e}$$
$$\mathbf{P}_{\mathrm{k|k}} = (\mathbf{I} - \mathbf{K}_\mathrm{k}\mathbf{H})\,\mathbf{P}_{\mathrm{k|k\text{-}1}} \tag{15f}$$

where $\hat{\mathbf{x}}_\mathrm{k} \in \mathbb{R}^{4\mathrm{N}+4}$ is the estimate of $\mathbf{x}_\mathrm{k}$, $\mathbf{P}_\mathrm{k} \in \mathbb{R}^{(4\mathrm{N}+4)\,\times\,(4\mathrm{N}+4)}$ is the estimate covariance matrix, $\mathbf{Q}_\mathrm{k} \in \mathbb{R}^{(4\mathrm{N}+4)\,\times\,(4\mathrm{N}+4)}$ is the process noise matrix, $\mathbf{K}_\mathrm{k} \in \mathbb{R}^{(4\mathrm{N}+4)\,\times\,(2\mathrm{N}+2)}$ is the optimal Kalman gain, and $\mathbf{R}_\mathrm{k} \in \mathbb{R}^{(2\mathrm{N}+2)\,\times\,(2\mathrm{N}+2)}$ is the measurement noise matrix.

2) Resetting $\mathbf{x}_k$: As DFs are lost and new inserted on a regular basis in the image processing, handling logic is needed to reset the state estimation accordingly. The reason for this is to avoid infeasible motion transients in the DF position and velocity estimates. Thus, when a new DF is lost and a new added, the following reset is applied before calculating (15),

$$\mathbf{x}_{\mathrm{k\text{-}1|k\text{-}1}} = \mathbf{M}_\mathrm{k}\mathbf{x}_{\mathrm{k\text{-}1|k\text{-}1}} + (\mathbf{I} - \mathbf{M}_\mathrm{k})\boldsymbol{\beta}_\mathrm{k} \tag{16a}$$
$$\mathbf{P}_{\mathrm{k\text{-}1|k\text{-}1}} = \mathbf{M}_\mathrm{k}\mathbf{P}_{\mathrm{k\text{-}1|k\text{-}1}}\mathbf{M}_\mathrm{k} + (\mathbf{I} - \mathbf{M}_\mathrm{k})\mathbf{P}_0 \tag{16b}$$
$$\mathbf{Q}_\mathrm{k} = \mathbf{M}_\mathrm{k}\mathbf{Q}_0 + \kappa_Q(\mathbf{I} - \mathbf{M}_\mathrm{k})\mathbf{Q}_0 \tag{16c}$$
$$\mathbf{R}_\mathrm{k} = \mathbf{M}_\mathrm{k}\mathbf{R}_0 + \kappa_R(\mathbf{I} - \mathbf{M}_\mathrm{k})\mathbf{R}_0 \tag{16d}$$

where $\mathbf{M}_\mathrm{k} \in \mathbb{R}^{(4\mathrm{N}+4)\,\times\,(4\mathrm{N}+4)}$ is a diagonal matrix for which the elements are set according to

$$M_{ii} = \begin{cases} 0 & \text{if new DF} \\ 1 & \text{else.} \end{cases} \tag{17}$$

Moreover, $\mathbf{I} \in \mathbb{R}^{(4\mathrm{N}+4)\,\times\,(4\mathrm{N}+4)}$ is an identity matrix, and $\boldsymbol{\beta} \in \mathbb{R}^{4\mathrm{N}+4}$ is a reset vector composed by the N DF vectors (and the vessel position and velocity, as $\mathbf{x}$), which is set by

$$\boldsymbol{\beta}_i = \begin{cases} \begin{bmatrix} \mathbf{y}_k^d + \mathbf{y}_k^s - h\hat{\mathbf{v}}_{ice} \\ \hat{\mathbf{v}}_{ice} \end{bmatrix} & \text{if new DF} \\ 0 \quad \text{else,} \end{cases} \tag{18}$$

where $\boldsymbol{\beta}_i \in \mathbb{R}^4$. Furthermore, $\mathbf{P}_0$, $\mathbf{Q}_0$, and $\mathbf{R}_0$ are initial covariance, process, and noise matrices, respectively, and κ_Q and κ_R are tunable gains to manipulate the estimated DF behavior of the first iteration after a reset.

3) Signal selection: The objective of the signal selection in Figure 3 is to remove DFs that do not originate from the ice cover around the vessel. This is done by calculating the error norm of the individual DF velocities with respect to the global estimated drift and using this to determine $\mathbf{C}_\mathrm{k}$ in (8). The error norm $e_\mathrm{k}^{\mathrm{DF}_i} \in \mathbb{R}$ is defined as

$$e_\mathrm{k}^{\mathrm{DF}_i} = \sqrt{\|\hat{\mathbf{v}}_\mathrm{k}^\mathrm{d} - \hat{\mathbf{v}}_{\mathrm{k\text{-}1}}^\mathrm{ice}\|_2}, \quad \forall i \in [1, N]. \tag{19}$$

To compute $\mathbf{C}_\mathrm{k}$, we define

$$\bar{\mathbf{C}}_\mathrm{k} = \begin{bmatrix} 0 & \mathbf{C}_\mathrm{k}^{\mathrm{DF}_1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{C}_\mathrm{k}^{\mathrm{DF}_2} & \cdots & 0 & 0 & 0 \\ \vdots & & & & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \mathbf{C}_\mathrm{k}^{\mathrm{DF}_N} & 0 & 0 \end{bmatrix}, \tag{20}$$

where $\mathbf{C}_\mathrm{k}^{\mathrm{DF}_i} \in \mathbb{R}^{2\times 2}$ for $i = 1, 2, ..., N$ are unit selection matrices. These are determined by

$$\mathbf{C}_\mathrm{k}^{\mathrm{DF}_i} = \begin{cases} \mathbf{I} & \text{if } \; e_k^{DF_i} \in \mathcal{D}_k \cup \mathcal{D}_k^{min} \\ 0 & \text{else.} \end{cases} \tag{21}$$

We define here

$$\mathcal{D}_\mathrm{k} := \{e_\mathrm{k}^{\mathrm{DF}_i} \leq e_M\} \tag{22a}$$
$$\mathcal{D}_\mathrm{k}^{\min} := \min_{\kappa_e}(e_\mathrm{k}^{\mathrm{DF}_1}, e_\mathrm{k}^{\mathrm{DF}_2}, \ldots, e_\mathrm{k}^{\mathrm{DF}_N}), \tag{22b}$$

where $e_M \in \mathbb{R}_{>0}$ is a threshold value such that $\mathcal{D}_\mathrm{k}$ is the set $e_\mathrm{k}^{\mathrm{DF}_i}$ values where $e_\mathrm{k}^{\mathrm{DF}_i} \leq e_M$ holds. $\mathcal{D}_\mathrm{k}^{\min}$ is the set of the κ_e number of smallest e_k^{DF} values. The objective of $\mathcal{D}_\mathrm{k}^{\min}$ is to provide a minimum number of DFs to the ice-drift estimator. Finally, $\mathbf{C}_k$ is computed from $\bar{\mathbf{C}}_k$ by removing its zero-rows.

Notice that this will effectively vote out DFs originating from objects that are moving significantly faster than the ice-drift. Examples of such include other ships, helicopters, and some types of moving overlay graphics. These are then removed from application in the ice-drift estimator. However, non-ice DFs that are in $\mathcal{D}_\mathrm{k} \cup \mathcal{D}_\mathrm{k}^{\min}$ will be applied as measurements. Although no mechanisms are introduced to handle such, practice has shown that the selection of such DFs are few, and their implication on the final estimate is minimal. Therefore, no further action is taken to handle them in this paper. If knowledge of positions of specific non-ice structures are available, these can be removed by cropping or weighing out part of the image in the Prepare image algorithm.

4) The ice-drift estimator: A linear KF, as seen in (15), cannot be used to estimate $\mathbf{z}_\mathrm{k}$ since (8) is nonlinear. To deal with this we choose an Unscented Kalman filter (UKF) instead. This allows the nonlinearities of (6) to be implemented directly without linearization. The UKF employs a deterministic sampling technique to pick a minimal set of sample points, known as sigma points, to estimate the state and covariance matrix. Below is given a summary of the key equations. For more in-depth details the reader is advised to [14]. The applied UKF algorithm can be written as,

$$\hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}}^{(i)} = \mathbf{G}(\hat{\mathbf{z}}_{\mathrm{k\text{-}1|k\text{-}1}}^{(i)}) \tag{23a}$$
$$\hat{\mathbf{z}}_{\mathrm{k\text{-}1|k\text{-}1}}^{(i)} = \hat{\mathbf{z}}_{\mathrm{k\text{-}1|k\text{-}1}} + \tilde{\mathbf{z}}^{(i)} \qquad \forall i = 1, ..., 2\epsilon \tag{23b}$$
$$\tilde{\mathbf{z}}^{(i)} = \left(\sqrt{\epsilon\mathbf{P}_{\mathrm{k\text{-}1|k\text{-}1}}^{\mathrm{ice}}}\right)_i^\top \qquad \forall i = 1, ..., \epsilon. \tag{23c}$$
$$\tilde{\mathbf{z}}^{(\epsilon+i)} = -\left(\sqrt{\epsilon\mathbf{P}_{\mathrm{k\text{-}1|k\text{-}1}}^{\mathrm{ice}}}\right)_i^\top \quad \forall i = 1, ..., \epsilon \tag{23d}$$
$$\hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}} = \frac{1}{2\epsilon}\sum_{i=1}^{2\epsilon}\hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}}^{(i)} \tag{23e}$$
$$\mathbf{P}_{\mathrm{k|k\text{-}1}}^{\mathrm{ice}} = \frac{1}{2\epsilon}\sum_{i=1}^{2\epsilon}(\hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}}^{(i)} - \hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}})(\hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}}^{(i)} - \hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}})^\top + \mathbf{Q}^{ice} \tag{23f}$$
$$\hat{\mathbf{z}}_\mathrm{k} = \hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}} + \mathbf{K}_\mathrm{k}^{\mathrm{ice}}(\mathbf{y}_\mathrm{k}^{\mathrm{ice}} - \hat{\mathbf{y}}_\mathrm{k}^{\mathrm{ice}}) \tag{23g}$$
$$\mathbf{P}_\mathrm{k}^{\mathrm{ice}} = \mathbf{P}_{\mathrm{k|k\text{-}1}}^{\mathrm{ice}} - \mathbf{K}_\mathrm{k}^{\mathrm{ice}}\mathbf{P}_\mathrm{y}\mathbf{K}_\mathrm{k}^{\mathrm{ice}\top} \tag{23h}$$
$$\mathbf{K}_\mathrm{k}^{\mathrm{ice}} = \mathbf{P}_{\mathrm{xy}}\mathbf{P}_\mathrm{y}^{-1}, \tag{23i}$$

Fig. 6. The icebreakers Oden and Frej in Longyearbyen September 16th 2015, before embarking on the OATRC15 research cruise. Picture: of Thomas W. Johansen.

where $\hat{\mathbf{z}}_{\mathrm{k}} \in \mathbb{R}^{10}$ is the estimate of $\mathbf{z}_{\mathrm{k}}$, the superscript (i) notation denotes one of the 2ϵ sigma points, ϵ is the number of elements in $\mathbf{z}$, $\mathbf{P}_{\mathrm{k}}^{\mathrm{ice}}$ is the covariance matrix, $\mathbf{Q}^{\mathrm{ice}}$ is the process noise matrix, $y_{\mathrm{k}}^{\mathrm{ice}}$ is defined in (8), and

$$\mathbf{P}_{\mathrm{xy}} = \frac{1}{2\epsilon}\sum_{i=1}^{2\epsilon}(\hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}}^{(i)} - \hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}})(\hat{\mathbf{y}}_{\mathrm{k}}^{(i)} - \hat{\mathbf{y}}_{\mathrm{k}})^{\top} \tag{24a}$$

$$\mathbf{P}_{\mathrm{y}} = \frac{1}{2\epsilon}\sum_{i=1}^{2\epsilon}(\hat{\mathbf{y}}_{\mathrm{k|k\text{-}1}}^{(i)} - \hat{\mathbf{y}}_{\mathrm{k|k\text{-}1}})(\hat{\mathbf{y}}_{\mathrm{k|k\text{-}1}}^{(i)} - \hat{\mathbf{y}}_{\mathrm{k|k\text{-}1}})^{\top} + \mathbf{R}^{\mathrm{ice}} \tag{24b}$$

$$\hat{\mathbf{y}}_{\mathrm{k}} = \frac{1}{2\epsilon}\sum_{i=1}^{2\epsilon}\hat{\mathbf{y}}_{\mathrm{k}}^{(i)} \tag{24c}$$

$$\hat{\mathbf{y}}_{\mathrm{k}}^{(i)} = \mathbf{C}_{\mathrm{k}}^{\mathrm{ice}}\hat{\mathbf{z}}_{\mathrm{k|k\text{-}1}}^{(i)} \qquad \forall i = 1, ..., 2\epsilon, \tag{24d}$$

where $\mathbf{R}^{\mathrm{ice}}$ is the process noise matrix. Since the UKF applies the nonlinear model (8), which incorporates the cyclic behavior often seen in drifting sea-ice, it may be used to predict the ice-drift some hours ahead in time. This is done by running the filter in dead reckoning. However, for the prediction to be accurate, the estimates of $\mathbf{a}$, $\boldsymbol{\omega}$,ϕ and $\mathbf{b}$ must be given time to converge. Also, care must be taken as the prediction accuracy will deteriorate with the length of the prediction horizon.

IV. Experimental results

As mentioned in the introduction, in September 2015 a 14-day research expedition, OATRC15, was carried out north of Svalbard. It was a two-ship operation using the Swedish icebreakers Oden and Frej (shown in Figure 6) for studies on IM and ship performance in ice. The expedition was supported by the ExxonMobil Upstream Research Company and performed by the Norwegian University of Science and Technology (NTNU) in cooperation with the Swedish Polar Research Secretariat and the Swedish Maritime Administration.

A. Setup and data logging

During the cruise, the radar images from Oden's Consilium Selesmar Selux ST-340 WS ARPA display was logged at 1 Hz using an Epiphan USB2DVI 3.0 video grabber unit. The radar system can be set up to use both S-band and X-band, but only X-band was used in the experiments. More details about the radar system can be found in [15]. The Epiphan frame grabber was connected to the video signal of an auxiliary monitor on the bridge using a VGA splitter and to a laptop running Windows 7 using a USB 3.0 cable. The laptop's clock was synchronized with an onboard time server, and a proprietary Epiphan frame grabbing software was used to save and timestamp the radar images. The position data of the vessel was available as a part of an NMEA datasteam acquired via Ethernet network (along with NMEA data from other sensors). The complete datastream was logged by the same laptop logging the radar images. For reference, all variables were recorded with respect to coordinated universal time (UTC).

As a part of the IM operation, multiple ice-drift beacons were deployed and retrieved by a helicopter to suitable ice floes in the surroundings of the operation area. These are applied as validation of the proposed algorithm.

It must be emphasized that the estimation algorithm was not run online during the cruise, but the radar images and position data were logged and stored for convenient desktop reconstruction of the data-streams for algorithm development and post-processing in Matlab.

B. Ice conditions and operation

The data presented in this paper is a 28 hour data segment starting at 10:40 on 25th September 2015 and ends at 14:50 on the 25th September 2015. The ice conditions in the interval were monitored by an experienced ice observer crew member to vary between 9/10 and 10/10 ice concentration with ice floes ranging from 500 m across to brash (predominantly in the range 500-50 m). The visibility from the Oden bridge ranged from 1 km to 10 km as fog and snow was occasionally present. The wind direction varied between 308° and 2° (0° is true north) with wind speed between 4.0 m/s to 6.7 m/s.

In Figure 7, both the traces of Oden and one beacon from this time-segment can be seen along with the operation area. The beacon track shows a general south-west ice-drift trend that contains significant drift curvature and velocity changes. The Oden position track shows heavy maneuvering as a part of the IM operation. In general, the objective was to break up the incoming ice so that the ice floe distribution hitting a geographic position downstream of the IM operation was reduced. As a part of the normal operation, the Oden crew occasionally made adjustments to the radar display. Table I summarizes these interaction events. Since radar images originate from the display of operator station, all interaction from the operator is embedded in the captured data.

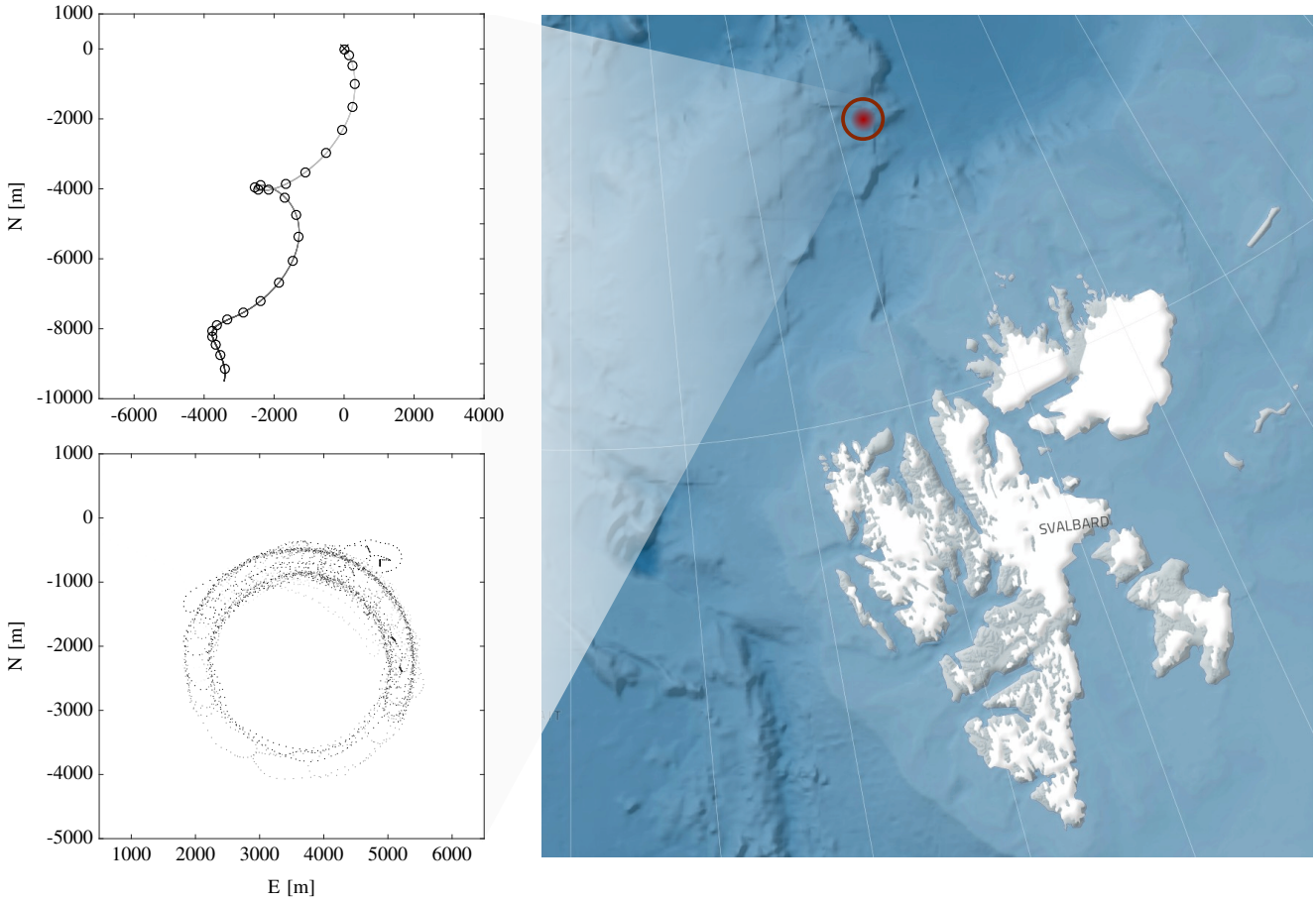

Fig. 7. The map shows the operation area of the OATRC15. The plot in the top left corner shows the ice-drift as recorded by an ice-drift beacon from the 28 hour data segment considered in this paper. The circles mark hour intervals. The plot in the lower left corner shows the position trace of Oden during the same interval.

TABLE I
RADAR INTERACTION BY CREW OPERATORS.

Time	Comment
20:19	Radar range set from 3 to 1.5 nmi
20:53	Radar gain reduced from 89% to 82%
20:53	Radar range set from 1.5 to 0.75 nmi
20:53	Radar range set from 0.75 to 1.5 nmi
21:18	Radar range set from 1.5 to 3 nmi
21:18	Radar range set from 3 to 0.75 nmi
21:18	Radar range set from 0.75 to 1.5 nmi
21:48	Radar view set to off-center PPI
21:48	Radar range set from 1.5 to 0.75 nmi
21:48	Radar range set from 0.75 to 1.5 nmi
21:48	Radar view set to centered PPI

C. Algorithm setup and parameters

As mentioned, the ice-drift estimation algorithm was tested by an implementation in Matlab. The algorithm parameters were set partly with signal analysis of the measurement and partly by trial and error. Table II provides the values applied in the image processing, and Table III provides the values applied in the covariance resetting and the signal selection. Tables IV and V provide the diagonal terms in $\mathbf{R}_0$, $\mathbf{Q}_0$, $\mathbf{R}^{\text{ice}}$, and $\mathbf{Q}^{\text{ice}}$. For both KFs, the initial covariance matrices, $\mathbf{P}_0$ and $\mathbf{P}_0^{\text{ice}}$, were set to identity. For the ice-drift estimator, the initial state $\hat{\mathbf{z}}$ was set to

$$\mathbf{v}_0^{\text{ice}} = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top \tag{25a}$$
$$\mathbf{a}_0 = \begin{bmatrix} 0.12 & 0.115 \end{bmatrix}^\top \tag{25b}$$
$$\boldsymbol{\omega}_0 = 10^{-4} \cdot \begin{bmatrix} 1.4 & 1.45 \end{bmatrix}^\top \tag{25c}$$
$$\boldsymbol{\phi}_0 = \begin{bmatrix} 2.35 & 0.85 \end{bmatrix}^\top \tag{25d}$$
$$\mathbf{b}_0 = \begin{bmatrix} -0.1 & -0.06 \end{bmatrix}^\top . \tag{25e}$$

As seen in Table I the radar range and radar gain were adjusted several times during the presented datasegment. This implies that the radar display goes black before being filled with data again, and thus makes the DF measurements unreliable until the display is filled with consistent data again. To avoid unfeasible transients in the estimated variables, the DF motion KF is run in dead reckoning for 1 minute at each change. This allows the image processing to regain consistent DF measurements before they are allowed to affect the state estimates.

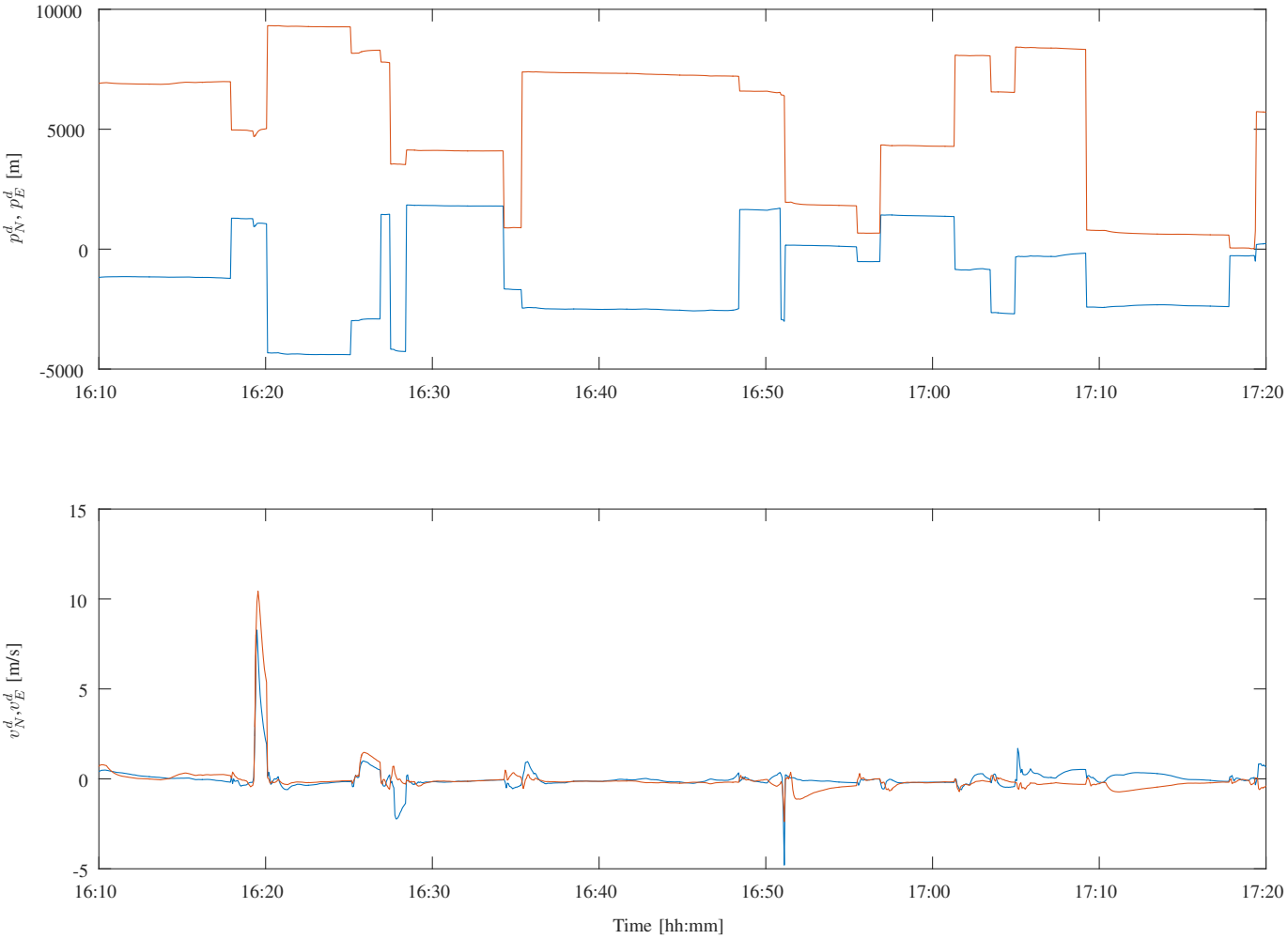

Fig. 8. The estimated position and velocity signals of one channel in the DF motion estimator.

TABLE II
PARAMETERS APPLIED IN THE IMAGE PROCESSING AND THE COMPUTATION OF $\mathbf{y}_k$.

Param.	Comment	Value
h	Step time, [s]	5
N	DF count	100
q_x	Pixel count north	723
q_y	Pixel count east	1106
θ_x	Center removal weight	$6.6 \cdot 10^{-5}$
θ_y	Center removal weight	$2 \cdot 10^{-4}$
q_w	Window side length	21
κ_b	$\mathbf{W}_b$ threshold	0.5
q_Λ	Window side length	101
κ_γ	$\mathbf{\Omega}$ mean threshold	5
C_R	$\mathbf{y}_k$ coefficient	$\frac{3704}{1121}$
R_r	Radar range [nmi]	3/1.5/0.75

TABLE III
PARAMETERS APPLIED IN THE MOTION ESTIMATOR RESETTING AND THE SIGNAL SELECTION.

Param.	Comment	Value
κ_Q	Process noise reset	1000
κ_R	Measurement noise reset	0.01
e_M	Acceptance threshold	0.0514
κ_e	Minimum numer of DFs	15

TABLE IV
MEASUREMENT NOISE VALUES APPLIED IN $\mathbf{R}_0$ AND $\mathbf{R}^{\text{ICE}}$. EQUAL VALUES WERE APPLIED IN THE NORTH AND EAST DOFS.

Param.	Comment	Value
r_{p}^{s}	GPS measurement noise	0.0275
r_{p}^{DF}	DF measurement noise	11
$r_{\text{v}}^{\text{ice}}$	DF velocity noise	0.02

D. Results and discussion

Figure 8 shows a one hour and ten minutes segment for one of the hundred tracked DFs of the DF motion estimator. The top plot shows the estimated position in the NED frame, where it can be seen that the algorithm changes the tracked DF point relatively often. The main reasons for this behavior is the heavy maneuvering of the vessel during IM, which causes the radar blind zone (seen in Figure 2) to swipe around the PPI on an irregular basis (causing loss of DFs). In the lower plot of Figure 8, the estimated DF velocity in the NED frame is shown.

It can be seen that significant velocity transients occur oc-

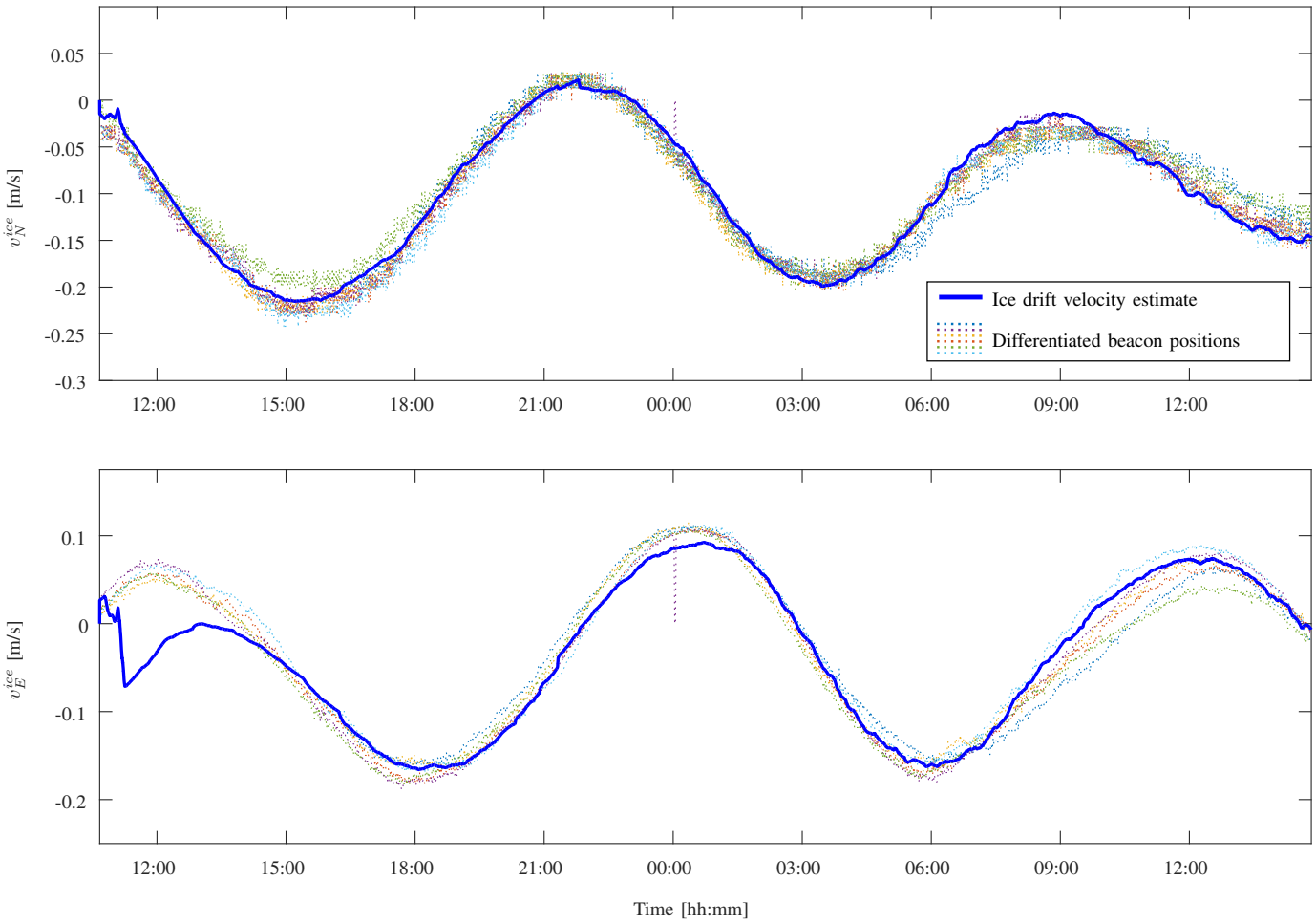

Fig. 9. Estimated ice-drift velocity compared to differentiated GPS signals.

TABLE V
PROCESS NOISE VALUES APPLIED IN $\mathbf{Q}_0$ AND $\mathbf{Q}^{\text{ICE}}$. EQUAL VALUES WERE APPLIED IN THE NORTH AND EAST DOFS.

Param.	Comment	Value
q_{p}^{s}	Vessel position	100
q_{v}^{s}	Vessel velocity	10
q_{p}^{DF}	DF position	0.001
q_{v}^{DF}	DF velocity	10^{-6}
$q_{\text{v}}^{\text{ice}}$	Ice velocity	$0.5 \cdot 10^{-7}$
$q_{\text{a}}^{\text{ice}}$	Signal amplitude	$0.5 \cdot 10^{-8}$
q_{ω}^{ice}	Signal frequency	$5 \cdot 10^{-16}$
q_{ϕ}^{ice}	Signal phase	$2.5 \cdot 10^{-9}$
$q_{\text{b}}^{\text{ice}}$	Signal offset	$0.5 \cdot 10^{-9}$

casionally after an estimator reset. Although convergence to reasonable ice-drift velocity values is relatively fast, and well handled by the signal selection, the behavior is unphysical. It is not critical to the performance, but the transients may be further improved by an alternative resetting mechanism, and possibly by introducing a velocity constraint in the DF motion estimator.

Figures 9 and 10 show the final ice-drift velocity estimate and the UKF parameter estimates, respectively. In Figure 9 the estimated ice-drift velocity vector is compared to the differentiated position signals of six ice-drift beacons physically placed on various ice floes in an area around the vessel. The fact that the beacons have a large spatial distribution may explain the spread in the measured velocity signal. Nevertheless, after the ice-drift parameter estimates have converged (at approximately 13:00) the correspondence between the differentiated GPS signals and the estimated is high. This indicates that the proposed system is providing reasonable estimates based on the radar pictures and GPS measurements alone. However, more data are needed to fully validate the algorithm as the convergence of the parameter estimates in $\hat{\mathbf{z}}$ must be evaluated further. The current dataset offers only about two periods of the ice-drift signal (as seen in Figure 9). This is insufficient to conclude on the parameter convergence. Yet, the results are encouraging. Alternatively, to avoid parameter estimation as presented in this paper, a linear KF may be used instead of the UKF as the ice-drift estimator. The major downside of that is the loss of prediction capability including the cyclic behavior, since the nonlinear model in (8) cannot be applied.

In general, the presented results indicate that remote sensing of the operationally important ice-drift velocity vector can be achieved by onboard sensors alone. The algorithm's incorporation of the nonlinear model (8) and a UKF also allow for prediction of the ice-drift sinusoidal behavior. This

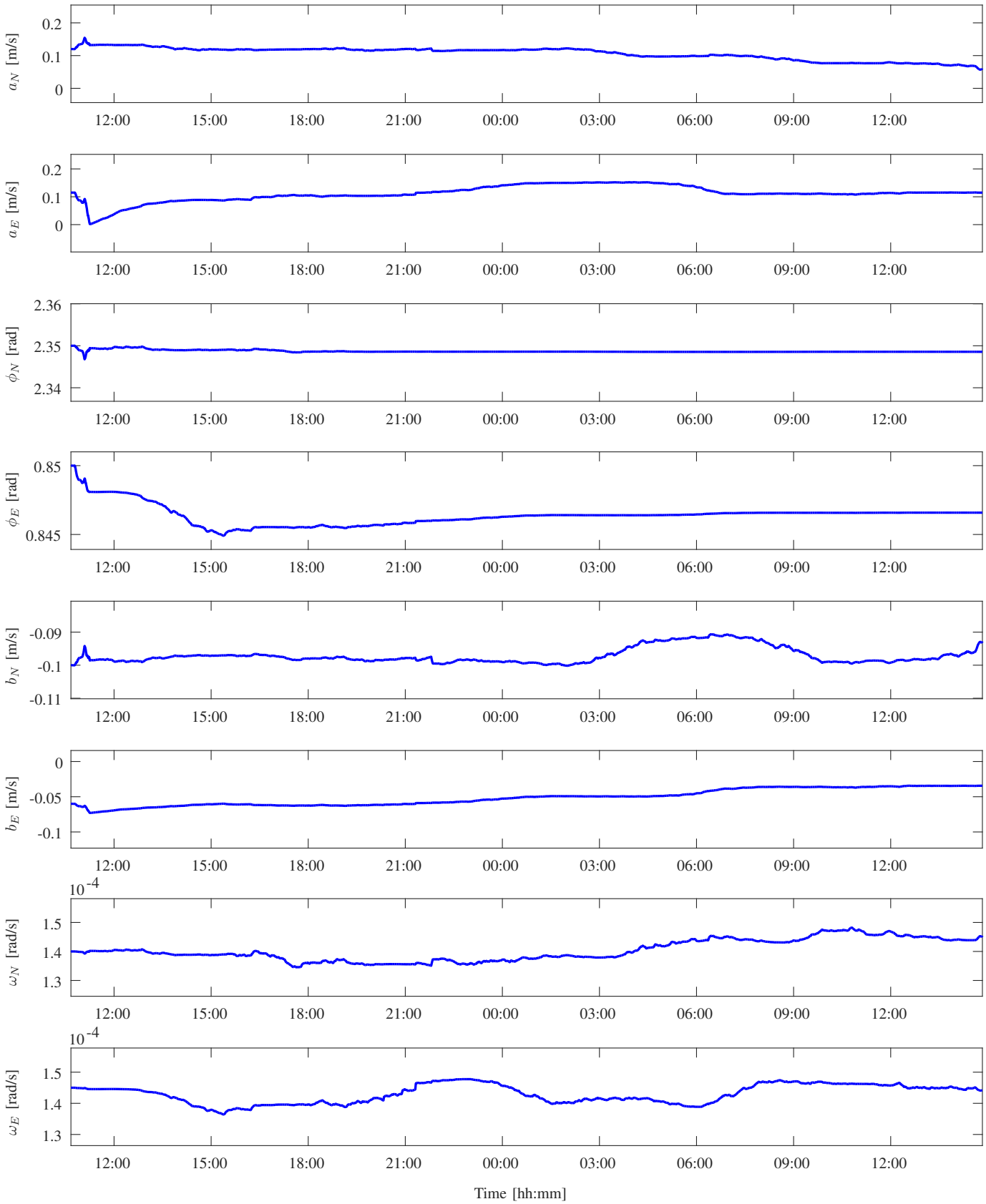

Fig. 10. Paramter estimates of the ice-drift velcity signal from the UKF.

is operationally very important for decision support and planning. However, the prediction capability is not studied further in this paper as it requires more validation data. The implication of the work is that the use of ice-drift beacons

may be significantly reduced. That is beneficial, as fewer helicopter operations are required (which reduce the risk and environmental impact). Also, it offers ice-drift velocity monitoring when helicopter operations are not possible (in case of precipitation, fog, and other no-fly conditions).

It is also worth mentioning that the processing time in Matlab per iteration of the algorithm was about 1 second. That suggest that suitable real-time capability when implemented in a more efficient programming language, can be achieved.

The sensor fusion nature of the KFs implies a potential for further integration with other sensor technologies to provide an enhanced operational decision support tool for surveillance, detection, monitoring, and forecasting of the physical environment. Examples of extensions that may improve the presented performance is:

- Incorporation of further vessel specific measurements and models to improve the accuracy of the vessel state estimation in the DF motion estimator. See [16] for potential models and their required measurements.
- Apply radar data with higher resolution that have not been processed for display to an operator.
- Improve (2) by choosing another integration method than forward Euler.

Finally, this work may be enhanced and extended to form an operational monitoring, detection, and surveillance tool by incorporating different sensing technologies. Examples include physical ice-drift beacons for special high risk hazards, wind sensors, satellite images, computer vision systems, and LIDAR systems.

V. Conclusions

This paper presents a novel remote sensing algorithm capable of estimating the ice-drift velocity vector with comparable accuracy of differentiating GNSS measurements. This was achieved by detecting and tracking distinctive ice features in radar images through image processing, removing the vessel motion, and fusing multiple tracked points using two Kalman filters in cascade. The main implications for the work is that it may reduce helicopter operations, since ice-drift velocity monitoring need not to rely solely on physical beacons deployed on the ice cover.

VI. Acknowledgment

Research partly funded by the Research Council of Norway project no. 203471: CRI SAMCoT. Also, the authors would like to acknowledge ExxonMobil Upstream Research Company, the Swedish Polar Research Secretariat, and the Swedish Maritime Administration for their efforts in planning and carrying out OATRC15. Finally, the authors would like to thank the helpful crew of the icebreakers Oden and Frej.

References

[1] M. Leppäranta, *The Drift of Sea Ice*. Springer, 2011.

[2] A. Berg and L. E. B. Eriksson, "Investigation of a hybrid algorithm for sea ice drift measurements using synthetic aperture radar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 5023 – 5033, 2014.

[3] D. Power, P. Bobby, C. Howell, F. Ralph, and C. Randell, "State of the art in satellite surveillance of icebergs and sea ice," in *Proc. Offshore Technology Conference*, 2012.

[4] R. Yulmetov, S. Løset, and K. Eik, "Analysis of drift of sea ice and icebergs in the Greenland sea," in *Proc. POAC*, 2013.

[5] R. Yulmetov, A. Marchenko, and S. Løset, "Ice drift and sea current analysis in the northwestern Barents sea," in *Proc. POAC*, 2013.

[6] R. Pilkington, A. Keinonen, and I. Sheikin, "Ice Observations and Forecasting During the Arctic Coring Project, August - September 2004," in *Proc. ICETECH*, 2006.

[7] Rutter, "Rutter Ice Navigator Detection and Navigation system," http://www.rutter.ca/ice-navigator, 2016, [Online; accessed 19-10-2017].

[8] J. Karvonen, "Virtual radar ice buoys a method for measuring fine-scale sea ice drift," *The Cryosphere*, vol. 10, no. 1, pp. 29–42, 2016.

[9] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[10] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, no. 1, 1981, pp. 674–679.

[11] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994*, 1994, pp. 593–600.

[12] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," *Optics letters*, vol. 33, no. 2, pp. 156–158, 2008.

[13] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more*. Springer, 2008.

[14] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[15] Consilium, "Selemar Radars," http://www.indumarver.com/archivos/CONSILIUM/RADARS_Selesmar_Selux_Radar.pdf, 2008, [Online; accessed 19-10-2017].

[16] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.