ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/318132722

A Target Tracking System for ASV Collision Avoidance Based on the PDAF

Chapter *in* Lecture Notes in Control and Information Sciences · May 2017

DOI: 10.1007/978-3-319-55372-6_13

CITATIONS
2

READS
88

3 authors, including:

Edmund Brekke
Norwegian University of Science and Technology

47 PUBLICATIONS **159** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Autosea View project

All content following this page was uploaded by Edmund Brekke on 12 November 2018.

The user has requested enhancement of the downloaded file.

A Target Tracking System for ASV Collision Avoidance Based on the PDAF

Erik F. Wilthil, Andreas L. Flåten, Edmund F. Brekke

Abstract Safe navigation and guidance of an Autonomous Surface Vehicle (ASV) depends on automatic sensor fusion methods capable of discovering static and moving obstacles in the vicinity of the ASV. A key component in such a system is a method for target tracking. In this paper we report a complete radar tracking system based on the classical Probabilistic Data Association filter (PDAF). The tracking system is tested on real radar data recorded in Trondheimsfjorden, Norway.

1 Introduction

Target tracking is a key ingredient in Collision Avoidance (COLAV) for autonomous vehicles. It is important to discover moving and stationary obstacles in data streams from imaging sensors such as radar, sonar, lidar, camera etc. Target tracking is required for reliable detection of moving objects under low to moderate Signal to Noise Ratio (SNR), and the predictions of a tracking method can enable proactive ownship manoeuvers.

Erik F. Wilthil
Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, e-mail: erik.wilthil@itk.ntnu.no

Andreas L. Flåten
Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, e-mail: andreas.flaten@itk.ntnu.no

Edmund F. Brekke
Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, e-mail: edmund.brekke@itk.ntnu.no

Dozens of tracking methods, or more specifically data association methods, exist [1, 12, 17, 13]. All of these methods attempt to make a judgement regarding the origin of measurements from the imaging sensors, either by hedging over different association hypotheses, or by minimizing the risk of misassociations. The arguably simplest state-of-the art tracking method is the PDAF [1], which is a single-target tracking method based on hedging. While target tracking has been extensively studied in the literature from a military point of view, very few research papers (e.g., [15] and [18]) have provided in-depth discuss of sensor fusion for COLAV systems.

In this paper we report the development of a complete PDAF-based tracking system intended for usage in future COLAV experiments. This includes the pipeline consisting of plot extraction (detection and segmentation), masking of land returns, integration with the ASV navigation system, and track management (track initiation and termination). We use AIS data to determine suitable values of the process noise covariance. The tracking system is tested on real data recorded by an frequency-modulated continuous-wave (FMCW) X-band radar onboard the Telemetron ASV.

The remainder of document is divided into several sections. In Section 2 we describe the radar used and how the raw data from this sensor is preprocessed. Section 3 describes the PDAF, and our particular implementation including track management. Section 4 describes analysis of AIS data recorded as part of the experiments, while Section 5 is devoted to the experiments and analysis of the experimental results. In Section 6 we discuss some key topics with basis in the results, before a conclusion follows in Section 7.

2 Pre-Tracking Radar Pipeline

Interpreting radar data in order to track targets is a process that can be broken down into several stages. An overview of the radar processing pipeline that is presented in this section is illustrated in Figure 1.

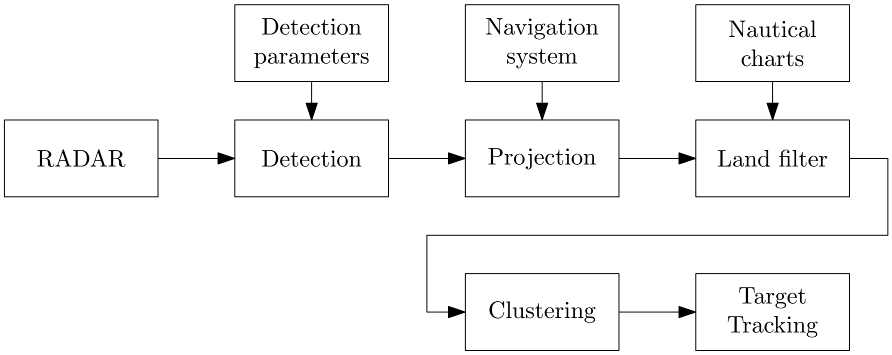

Fig. 1 An block diagram of the proposed radar tracking pipeline.

2.1 Detection

The tracking system is designed to work with the Navico 4G broadband radar, which has a proprietary detection process optimized for use at sea. The output of the detection stage in the radar processing pipeline is a stream of radar spokes, where each spoke contains detections for the range resolution cells for a given azimuth angle. This is illustrated in Figure 2.

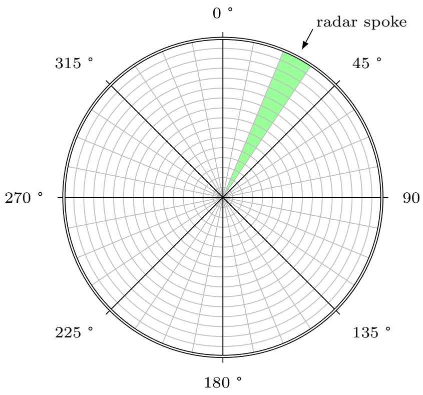

Fig. 2 An illustration of the resolution cells in a full radar scan, and a single radar spoke.

If a reliable in-built detection procedure is not provided by the radar, standard techniques based on the principle of Constant False Alarm Rate (CFAR) should be considered as the most straightforward approach. See [7].

2.2 Projection

As a next stage in the processing pipeline, we have chosen to transform the detections in to a world fixed reference frame. This mitigates distortion due to the effects of vehicle motion and rotating radar antenna. Since the platform is moving, this step depends on the ability to estimate the pose of the radar relative to a world fixed frame. It will be assumed that this transformation is perfectly known through a navigation system, and that the navigation system is synchronized in time with the radar. It is further assumed that due to the high vertical beam-width of the radar, detections can be interpreted directly in a horizontal reference frame (i.e. parallel to the nominal sea surface). The high beam-width also comes with a cost of distortion when the radar experiences wave induced motion.

Given a stream of radar spokes with known azimuth angle α, the first step is to transform a spoke from polar to Cartesian coordinates. A single detection can be represented in a horizontal Cartesian reference frame as:

$$p_i^h = \begin{bmatrix} \rho_i \cos\alpha \\ \rho_i \sin\alpha \\ 0 \end{bmatrix} \tag{1}$$

Where i is the index of the range resolution cell in the radar spoke, and ρ_i is the range of that resolution cell in meters. Given that the navigation system provides the translation r_{wh}^w and rotation R_h^w of the horizontal frame, we can express the detection in a world fixed frame as:

$$p_i^w = r_{wh}^w + R_h^w p_i^h \tag{2}$$

If the world fixed frame is a local North-East-Down (NED) frame, then this transformation can be approximated by a planar translation and a rotation about one axis.

The spokes are projected when they arrive, and aggregated into a point cloud made from a full rotation of the radar (a complete scan). The output of the projection stage is thus a world fixed two-dimensional point cloud of radar detections.

2.3 Land Masking

When land is within range of the radar, it receives a significant amount of returns, as seen in Figure 3. If these detections propagates further down the radar pipeline, they will introduce false tracks and extra computational load. To avoid this, we eliminate the detections on land by means of land masking. We use data obtained from The Norwegian Mapping Authority (Kartverket). The data is provided at a vector format describing the land area as a set of polygons. To use this efficiently, the map is preprocessed, and represented as a binary grid in the world-fixed frame the radar detections are transformed into. If the point lands in a cell containing land, the detection is discarded.

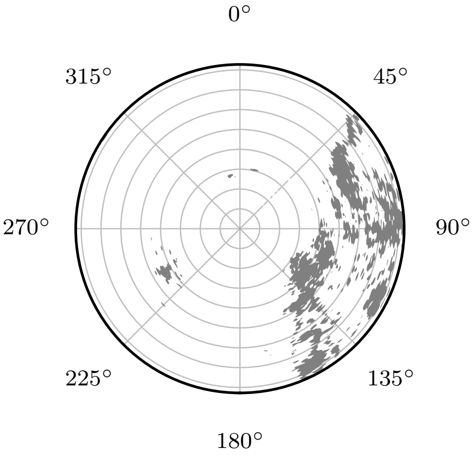

Fig. 3 The radar returns from land. The small dots in the middle of the plot shows returns from two ships. The points to the left shows the island of Munkholmen, while the large clusters to the right is Trondheim.

The primary error source in the map itself is the accuracy of the data. Kartverket does provide accurate data, but some offsets and reflections close to shore must be expected. We will see some effects of this in Section 5. Insufficiently accurate maps can be dealt with in several ways. One simple solution is to inflate the maps such that all detections within some distance to land are discarded as well. More advanced methods includes estimating the mismatch using, e.g., occupancy grids [16], but no such methods will be considered here.

2.4 Clustering

A standard assumption in target tracking algorithms is that a single target generates at most one measurement per scan. This assumption is useful in order to keep the tracking problem well-posed and tractable within the Bayesian framework. If this assumption is relaxed, then the Bayesian framework may need a model for the number of target measurements, which may be very difficult to specify.

However, this assumption will typically be violated when sensor resolution is high relative to the physical target extent. While the PDAF itself does not suffer very much from such violations, other tracking methods are much more sensitive to this assumption, and in any case having an abundance of measurements from the same target makes track initialization much more complicated.

Therefore, we attempt to uphold the at-most-one-measurement assumption by means of *clustering* of detections. A clustering algorithm assumes that detections that are in some sense close to each other originate from the same target, and join these detections together to form a cluster. Once a cluster is established, the detections in the cluster represent a single measurement. The location of the measurement can be computed as the centroid of the individual detections, weighted by the strength of the detections if amplitude information is available.

A simple definition of a cluster is a collection of cells where each cell neighbours at least one other cell in the cluster. The definition is not complete without the notion of a neighbouring point. A point p_i^w is defined as a neighbour to another point p_j^w if the Euclidian distance between the points is below some threshold. Mathematically this is defined as the binary function:

$$N_R(p_i^w, p_j^w) = \begin{cases} 1, & \|p_i^w - p_j^w\| \leq R \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

A set of points together with this function defines a graph $G = (V, E)$, where $V = \{1, 2, \ldots, N\}$ represent the point indices, and there is an edge from vertex i to j whenever (3) is equal to 1:

$$E = \left\{(i, j) \,|\, N_R(p_i^w, p_j^w) = 1,\ i, j \in V\right\} \tag{4}$$

Note that the number of edges in this graph depends on the value of R, and that a larger R implies a denser graph. Clusters can be computed as the *connected components* of G. Finding the connected components of a graph can be done by either a Depth-First Search (DFS) or Breadth-First Search (BFS). Constructing this graph has computational complexity $O(N^2)$ since we need to compute the Euclidian distance between each pair of points, while the graph search is $O(V + E)$ for both DFS and BFS. When there are thousands of detections, the $O(N^2)$ time required to build the graph can become prohibitively large. By representing the detected points in a *k-d tree* [6, 11], we can traverse this graph implicitly by performing radius searches at each vertex. Constructing the k-d tree has complexity $O(N \log N)$, and performing a radius search has an *average* complexity of $O(\log N)$. Traversing the graph in either BFS or DFS manner requires a radius search at each vertex, so the total average complexity (including building the tree) will be $O(N \log N)$.

3 PDAF-based Tracking Module

3.1 Motion Model

The state of a target is given as $x = [N, V_N, E, V_E]$, where N, E, V_N and V_E are the north and east positions and velocities of the target in a stationary NED reference frame. A discrete-time white noise acceleration model [9] is used to model the target motion. This model can be written on the form

$$x_{k+1} = F_T x_k + v_k \qquad\qquad p(v_k) = \mathcal{N}(v_k; 0, Q_T) \tag{5}$$

where v is process noise, $\mathcal{N}$ denotes the normal distribution and the state transition matrix F_T and noise covariance Q_T are given as

$$F_T = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q_T = \sigma_a^2 \begin{bmatrix} T^4/4 & T^3/2 & 0 & 0 \\ T^3/2 & T^2 & 0 & 0 \\ 0 & 0 & T^4/4 & T^3/2 \\ 0 & 0 & T^3/2 & T^2 \end{bmatrix} \tag{6}$$

where $T = t_{k+1} - t_k$ is the sample time, in general time-varying for a rotating radar. The process noise strength σ_a is selected according to the targets' expected maneuverability (See Section 5.3).

The measurements used in the PDAF are of target position only. Thanks to the Projection step described in Section 2.2, we can use a linear Cartesian measurement model of the form

$$z_k = H x_k + w_k \qquad\qquad p(w_k) = \mathcal{N}(w_k; 0, R) \tag{7}$$

where w is the measurement noise and H is given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{8}$$

We use a constant and diagonal measurement noise covariance matrix, reflecting the fact that target extent generally is larger than the size of the radar resolution cells (See Section 5.2).

3.2 Track Initiation

Tracks are formed using $2/2\&m/n$ logic [1]. A measurement that is not associated with any existing targets is a candidate for a new track, also called a *tentative* track. To estimate the state of a target, another measurement is required. This second measurement must fall within a circle-shaped validation gate around the tentative track. The radius of this circle is given by the maximum velocity of the target, the measurement noise statistics, and the probability of missing a target. The expressions are given in [2 p. 247].

3.3 PDAF Tracking

For preliminary and confirmed tracks which have a state estimate and a corresponding covariance, we employ the PDAF to track the targets. Starting from the third scan, the standard Kalman filter prediction equations are used to propagate the tracks

$$\hat{x}_{k|k-1} = F_T \hat{x}_{k-1|k-1} \tag{9}$$

$$P_{k|k-1} = F_T P_{k-1|k-1} F_T^T + Q_T \tag{10}$$

The predicted covariance is used to set up a measurement validation gate in order to reduce the number of measurements we need to consider for each target. Measurements that do not fall within any validation gate are used as candidates for new tracks. The gating process utilizes the predicted measurements and the corresponding covariance

$$\hat{z}_k = H\hat{x}_{k|k-1} \tag{11}$$

$$S_k = HP_{k|k-1}H^T + R \tag{12}$$

from which we can calculate the Normalized Innovation Squared (NIS) as

$$\text{NIS} = (z_k^i - \hat{z}_k)^T S_k (z_k^i - \hat{z}_k) \tag{13}$$

$$= \nu_k^{i\,T} S_k^{-1} \nu_k^i < \gamma_G \tag{14}$$

where γ is a threshold used to determine if the measurement should be associated with the target or not. The gating threshold is found from the inverse Cumulative Distribution Function (CDF) of the χ^2-distribution with degrees of freedom corresponding to the dimension of the measurement. The value ν_k^i is called the *measurement innovation*.

After each target has been assigned a (possibly empty) set of measurements, the PDAF provides a moment-matched Gaussian approximation of the posterior of the target state by hedging on all the measurements in the association gate, as well as the possibility that no measurements may originate from the target. The probability of measurement z^i being the correct is given as

$$\beta_k^i = \begin{cases} \frac{1}{c} \exp\left(-\frac{1}{2} {\nu_k^i}^T S_k^{-1} \nu_k^i\right) & i = 1 \ldots m_k \\ \frac{1}{c} \frac{2(1-P_D P_G)}{\gamma} m_k & i = 0 \end{cases} \tag{15}$$

where c is a normalization constant. The posterior is then calculated as

$$\nu_k = \sum_{i=1}^{m_k} \beta_k^i \nu_k^i \tag{16}$$

$$K_k = P_{k|k-1} H^T S_k^{-1} \tag{17}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \nu_k \tag{18}$$

$$P_{k|k} = P_{k|k-1} - (1 - \beta_k^0) K_k H P_{k|k-1} + \tilde{P}_k \tag{19}$$

where

$$\tilde{P}_k = K_k \left(\sum_{i=1}^{m_k} \beta_k^i \nu_k^i {\nu_k^i}^T - \nu_k \nu_k \right) K_k^T \tag{20}$$

is called the Spread of Innovations (SOI). The posterior covariance estimate differs from the regular Kalman filter in two ways. The second term in Equation (19) is adjusted to account for the possibility of missing measurements, and the SOI inflates the covariance to account for the uncertainty in data association.

3.4 Track Management and Track Termination

As validation gates for tentative, preliminary and confirmed tracks can overlap and capture the same measurements, some form of assignment hierarchy is needed. Measurements will be assigned to confirmed, preliminary and tentative tracks in that order, such that the more established tracks take precedence over the newer tracks. For a given category, the measurement will be assigned to all tracks that gate it, for example two confirmed tracks. This hierarchy reduce the number of false tracks, but it does not provide any guarantees against track coalescence, i.e., that multiple tracks start to associate the same set of measurements, and converge to each other.

As a remedy for this is, we prune tracks by a similarity test [1]. Denote the distance between two tracks as $d = x^1 - x^2$. The same-target hypothesis H_0 is that $d = 0$, and the two estimates with means $\hat{x}^1$, $\hat{x}^2$ and covariances P^1, P^2 are estimates of the same target. To test this hypothesis, calcuate the mean $\hat{d}$ and covariance T of d as

$$\hat{d} = \hat{x}^1 - \hat{x}^2 \tag{21}$$

$$T = P^1 + P^2 - P^{12} - P^{21} \tag{22}$$

where P^{ij} is the cross-covariance of track i and j. This is nonzero because of the common process noise in the tracks under the same-target hypothesis. This value is impractical in a real-time tracking system and it has been approximated by the individual covariances P^1 and P^2 as per the technique described in [1]. With these values, we can accept the same-target hypothesis if

$$\gamma = \hat{d}^T T^{-1} d < \gamma_\alpha \tag{23}$$

where γ_α is a threshold such that

$$P(\gamma > \gamma_\alpha | H_0) = \alpha \tag{24}$$

Furthermore, any track which has failed to gate any measurements over the last 5 scans is terminated. This criterion terminates all tracks that go outside of the radar range.

4 AIS Filtering

In addition to maritime radar, the Automatic Identification System (AIS) is a core component of maritime collision avoidance. While most larger ships are required to have AIS onboard, it is not sufficient to rely on AIS information alone for collision avoidance since it is not universally adopted, unreliable, and may even be switched off on purpose. Further information about AIS and its reliability can be found in [8].

Nevertheless, most of the time AIS contains valuable collision avoidance information, and will in many cases compliment radar when detection is limited. In the ideal case, the AIS system can be thought of as a target tracking system without the problems induced by false alarms or misdetections. AIS data come in the form of structured messagest consisting of fields such as Maritime Mobile Service Identity (MMSI) transmitter identifier, Coordinated Universal Time (UTC) second time stamp, UTC second time of arrival, World Geodetic System (WGS84) latitude and logitude, Course Over Ground (COG) and Speed Over Ground (SOG) relative true north.

In this section we discuss how raw AIS messages can be processed in a probabilistic framework, to quantify the uncertainty of AIS messages, and to motivate the use of AIS as a compliment to radar target tracking and as a tool to validate target motion models.

4.1 Out-of-order message arrival

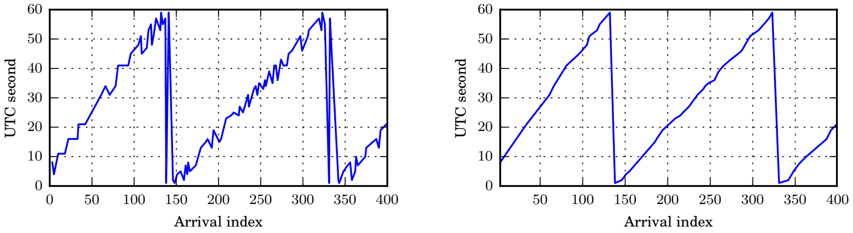

Fig. 4 Sequences of observed UTC second time stamps from the Trondheimsfjord II ferry, before and after filtering (left and right respectively). The "Arrival index" corresponds to the message arrival time. About 40% of the data is discarded in this case. Note that resulting sampling time is still on the order of a few seconds, implying that the loss of information is minimal.

One challenge in state estimation based on AIS is that AIS messages frequently arrive out-of-order, meaning that the UTC second time stamp is not locally monotonously increasing for a single MMSI. Optimal filtering will therefore require Out-Of-Sequence Measurements (OOSM) filtering methods [4]. An alternative to OOSM processing is to buffer messages and re-order them on the fly. This would add a time delay to the message arrival, which will propagate to target tracking and collision avoidance algorithms.

A simpler and more convenient solution is to sequentially throw away messages that defined as "old". While this clearly is a suboptimal solution, the information loss is not necessarily significant in practical terms. This effect can be compared to the information loss induced by reducing the sampling rate by a factor equal to the proportion of messages discarded, which is adequate if the resulting sampling time still is small enough to capture significant target dynamics.

A sequential time stamp filter can be implemented by storing the latest AIS message and comparing it to newly arriving messages. If an arrived message is newer than the currently stored message, it is assigned to be the latest message, else it is discarded. An illustration of this issue and the proposed solution can be seen in Figure 4, where the UTC time stamp does not always increase.

4.2 AIS Measurement Model

The AIS messages can be transformed into linear measurements of the kinematic state as defined in Section 3.1. Given a local stationary NED reference frame defined by a latitude, longitude and height over the WGS84 ellipsoid (equal to the local sea level), the position can be calulated according to formulas found in [5, p. 38]. The velocity in the same NED frame can be calculated as:

$$\begin{bmatrix} V_N \\ V_E \end{bmatrix} = \begin{bmatrix} V \cos \chi \\ V \sin \chi \end{bmatrix} \tag{25}$$

where V is the SOG and χ is the COG in the message. Since AIS is usually based on one or several Global Navigation Satellite Systems (GNSSs), a reasonable estimate of the measurement accuracy is conventional GNSS accuracy. Ideally the sampling time of the AIS messages should be exact, however if the actual sampling time is not an integer multiple of the UTC seconds, this will induce large position errors due to sampling time quantization. The SOG and COG estimates are less affected by this, since the course and speed usually do not change much within a single second. The actual error induced by quantization depends on the SOG and SOG of the vehicle, and how much the time stamp has been rounded off. The worst case rounding error can be assumed to be ± 0.5 seconds. To approximate this as an additional independent Gaussian error source, we add a position measurement covariance proportional to the speed in each direction (north or east), scaled by a suitable variance related to the quantization error. Since the actual quantization error is uniform in $(-0.5, 0.5)$ seconds, a reasonable choice is a moment matched Gaussian distribution with zero mean and standard deviation $\sigma_t = \frac{1}{\sqrt{12}}$. Using these definitions, the measurement matrix for the AIS measurements can be written as:

$$H = I_4 \tag{26}$$

and measurement covariance:

$$R_{AIS} = R_{GNSS} + \frac{1}{12} R_V, \quad R_V = \text{diag}\left[V_N^2, 0, V_E^2, 0\right] \tag{27}$$

Where the values used for R_{GNSS} are the following:

$$R_{GNSS} = \text{diag}\left[0.5^2, 0.1^2, 0.5^2, 0.1^2\right] \tag{28}$$

Note that since the measurement noise varies with speed, the measurement noise is time varying. An interesting observation is that with these values for R_{GNSS}, for speeds over approximately 2 metres per second, the discretization noise will dominate the position error.

4.3 Missing Measurement Kalman Filter

In addition to arriving out of sequence, AIS messages are recieved at varying rates. This is also the case for messages from a single ship, where the rate may vary significantly over time due to varying ship speed and and ship status [8]. It may also be the case that a message was discarded or not recieved at all. However, following the discussion in Section 4.2, it is convenient to run the AIS filter with a fixed sampling time of one second for all ships. When measurements are not available, the AIS filter simply uses Kalman filter prediction instead of Kalman filter update.

5 Experimental Validation

The complete tracking system has been validated on real data recorded during field trials with the Telemetron vehicle (owned by Maritime Robotics) in Trondheimsfjorden December 10 2015 and January 11 2016. The Telemetron vessel is equipped with a Navico 4G broadband radar, a commercial off-the-shelf AIS reciever and a Seapath 330+ high grade navigation system supplied by Kongsberg Seatex in Trondheim. The Robot Operating System (ROS) [14] software framework was used for data aquisition and algorithm implementation. The experiments involved two scenarios. In the first scenario, referred to as “Trondheimsfjord II”, Telemetron followed the catamaran Trondheimsfjord II, while also passing by several other ships. An overview is shown in Figure 9. In the second scenario, referred to as “Gunnerus”, Telemetron approached and encircled the research vessel Gunnerus, which lay stationary in the water east of Munkholmen, north of Trondheim harbor. An overview is shown in Figure 7.

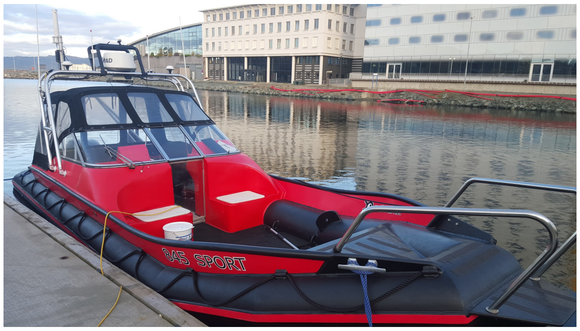

Fig. 5 The Telemetron ASV is equipped with several sensors such as a radar, AIS reciever, camera and navigation systems. For the purpose of this test, it was also equipped with a Kongsberg Seapath 330+ navigation system. The vehicle can both be used in autonomous mode and with manual control.

5.1 Clustering Runtime Experimental Analysis

The only computational bottleneck of significance in the tracking system is the clustering algorithm described in Section 2.4. In this section the runtime of the clustering algorithm is experimentally analyzed with data from the "Gunnerus" scenario, which is further elaborated in Section 5.2. The clustering algorithm was implemented in C++ and run on a laptop with a 2.8 GHz Intel Core i7 processor. The results of the runtime experiments can be seen in Figure 6. We see that the runtime increases both with respect to the number of detections per scan, and the clustering radius r. A lower clustering radius will in general lead to more and smaller clusters. This means that there is a tradeoff between being able to cluster fragmented detections and handling this further down in the processing pipeline. Even if real time implementation was not the main focus of this study, a critical remark is in order; the maximum rotation period for the radar used in these experiments is 2.5 seconds, implying that a runtime close to or above 2.5 seconds will mean that running this algorithm in a real time scenario will be infeasible. This "time limit" is often violated for a radius of 10 metres, and is also dangerously close for the other values of r. The clustering radius used in the rest of the experimental results is r = 5.0 metres.

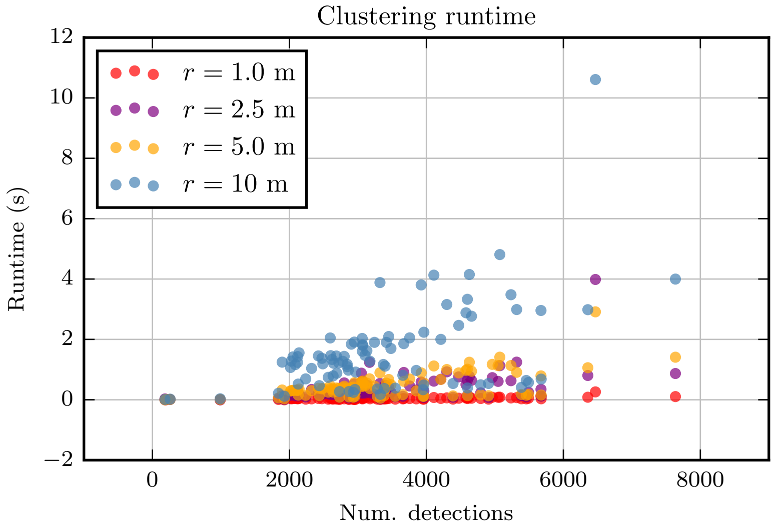

Fig. 6 Scatter plot of number of detection versus empirical runtime of the clustering algorithm described in Section 2.4.

5.2 Radar Measurement Noise Covariance Estimation

The determine a reasonable value of the RADAR measurement noise covariance, we analyzed the data from the "Gunnerus" scenario Figure 7. The ownship circled around Gunnerus for one and a half revolution, while recording radar data from Gunnerus. Since we know Gunnerus was stationary, the data from these recordings are convenient to use for estimation of the radar measurement noise covariance. Figure 8 shows the AIS position of Gunnerus, and the set of measurements within 100 meters of that position. These measurements forms the basis of our covariance estimation.

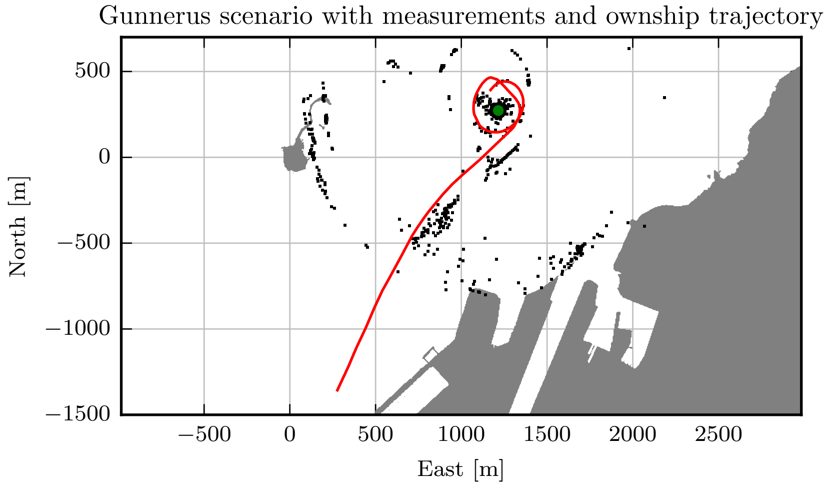

Fig. 7 The "Gunnerus" scenario. The red trajectory shows the ownship starting in the lower part of the image, and going north to circle around the NTNU research vessel Gunnerus, whose AIS position is shown as a green dot. The shaded areas shows the land, and the black dots are the cumulated set of measurements from this experiments. In addition to Gunnerus, the detections from a small boat can be seen passing on starboard side of the ownship.

By assuming that at most one detection per scan can originate from Gunnerus, we calculate the sample covariance as

$$\hat{R} = \frac{1}{K-1}\sum_{k=1}^{K}(z_{A_{\max}} - \bar{z}_{A_{\max}})(z_{A_{\max}} - \bar{z}_{A_{\max}})^T \tag{29}$$

where the measurement $z_{A_{\max}}$ is selected such that the convex hull of its cluster has the *largest area* within the 100 meter radius at this timestep. This is justified by assuming that the cluster returned from Gunnerus is larger than sea clutter and other

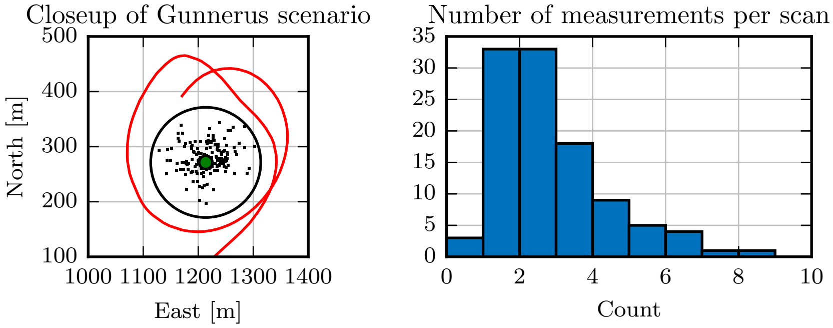

Fig. 8 253 measurements were found within 100m of Gunnerus at 107 timesteps. The histogram to the right shows the number of measurements per scan.

reflections. The difference between the average measurement and the average AIS position of Gunnerus is approximately 6.5 meters.

Notice that due to violations of the maximum-one-measurement assumption, it is not meaningful to use the rest of the measurements to calculate the clutter density of the scenario, as some tracking methods such as the Integrated Probabilistic Data Association (IPDA) [4] would do.

5.3 Determination of Process Noise Covariance

Given that the discrete white noise acceleration model presented in Section 3.1 is a realistic model, and that the measurement noise covariances have been determined, the only remaining parameter is the process noise variance σ_a^2. Having process and measurement noise parameters that are close to reality are crucial to performance, both in terms of Root Mean Squared Error (RMSE) and covariance consistency. Since the process noise parameter represents the assumed target maneuverability, the optimal choice of this parameter will likely vary somewhat between ships and even over time, especially with respect to the size of the ship. This section experimentally analyzes the choice of process noise variance for five ships in the vicinity of Telemetron over a 19 minutes time interval during the “Trondheimsfjord II” scenario, shown in Figure 9. Relevant characteristics based on AIS information from each vessel in the scenario is shown in Table 1.

The average NIS [2] was calculated as a measure of filter consistency, for a range of process noise variances. Note that the number of samples for each vessel differs, so that the two sided confidence interval for the NIS varies. Filter bias is also estimated by calculating the Average Innovation (AI) for all states, which should ideally be zero. The results are shown in Table 2. As emphasised by the bold font, it seems that the vessels can be grouped into two categories. Either they show little maneuverability, which corresponds to $\sigma_a = 0.05$, or they show significantly more maneuverability, where $\sigma_a = 0.5$ gives more appropriate values of the NIS. For

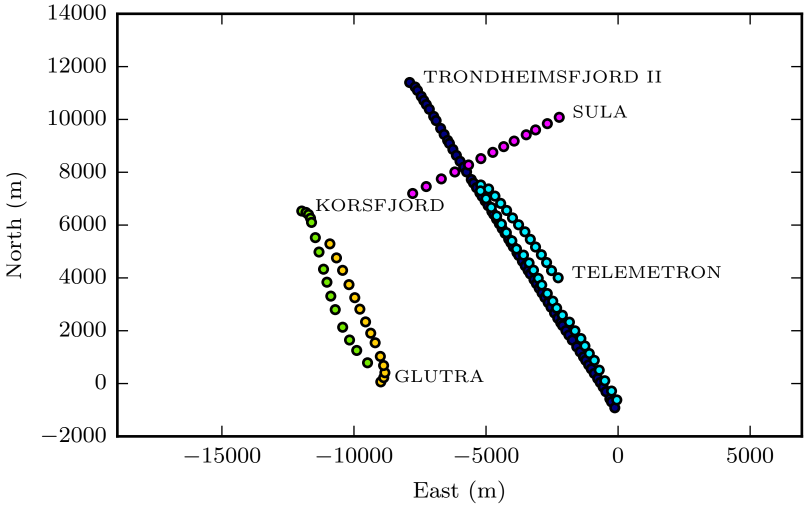

Fig. 9 An overview of the "Trondheimsfjord II" scenario. Only 1 out of 15 samples are shown for each trajectory, for visualization purposes.

Table 1 Vessel parameters in the "Trondheimsfjord II" scenario based on AIS information. Note the large diversity in ship type, size and mean SOG. The data was recorded on board the TELEMETRON vessel. KORSFJORD, GLUTRA and TRONDHEIMSFJORD II are passenger ferries that cross the fjord in Trondheim regularly.

Name	Type	Length x Breadth	Mean SOG
GLUTRA	Passenger	94.8 m x 16.0 m	5.3 m/s
SULA	Cargo	87.9 m x 12.8 m	5.7 m/s
KORSFJORD	Passenger	122 m x 16.7 m	5.6 m/s
TRONDHEIMSFJORD II	High speed	24.5 m x 8 m	12.8 m/s
TELEMETRON	Pleasure craft	8 m x 3 m	13.1 m/s

Table 2 Process noise evaluation via AIS filter consistency. The (r_1, r_2) interval is the two-sided 95% probability concentration region for the χ^2 distribution related to the corresponding NIS. This varies with according to the AIS data record length N. The NIS values that are closest to being covariance-consistent, i.e. closest to the 95% probability region, are emphasised in bold.

	$\sigma_a = 0.05$		$\sigma_a = 0.5$			
Name	NIS	AI	NIS	AI	(r_1, r_2)	N
GLUTRA	**4.67**	−0.02	0.90	0.01	$(3.47, 4.55)$	109
SULA	**3.61**	−0.22	0.51	−0.10	$(3.49, 4.56)$	106
KORSFJORD	71.8	−1.33	**4.31**	−0.44	$(3.52, 4.51)$	127
TR.FJORD II	11.3	−0.62	**3.24**	−0.16	$(3.76, 4.24)$	533
TELEMETRON	371	−0.04	**4.45**	−0.01	$(3.77, 4.23)$	579

simplicity it may be desirable to have a single value for the process noise variance. Choosing $\sigma_a = 0.5$ will thus in some cases give conservative values for the filter covariance, which presumably will inflict less risks of track-loss than the opposite choice. This value has been used in the following radar target tracking results.

5.4 Tracking Performance

We study the PDAF-based tracking system applied to the “Gunnerus” scenario in this section. The resulting tracks are shown in Figure 10.

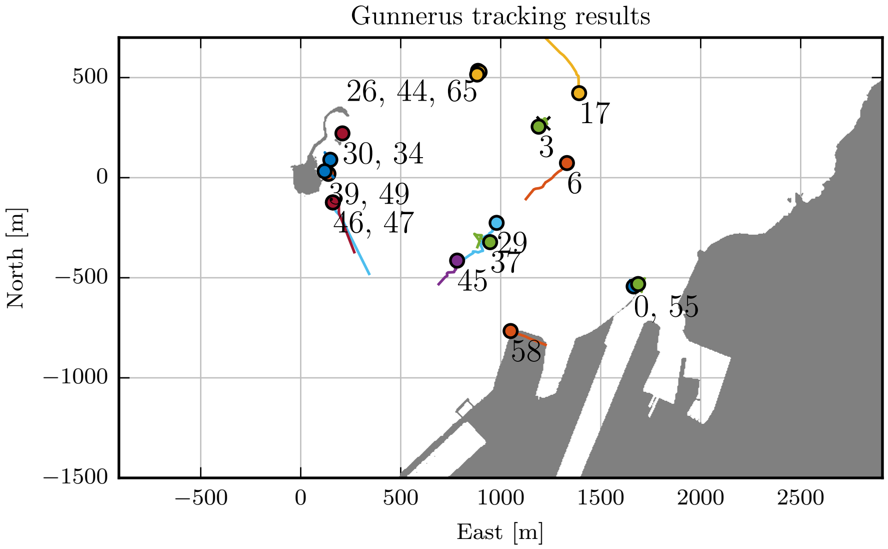

Fig. 10 Resulting tracks from the “Gunnerus” scenario. The circles mark the start of the tracks. The track labels are numbered in the order they are initiated. There are a total of 18 confirmed tracks.

First, observe that many of the tracks originate from land. The tracks stay close to or on land as long as the feature is in range of the radar. When the feature is outside of the radar range, due to the movement of Telemetron or radar range adjustments, some tracks start to move out into open sea. Since they are out of range of the radar, no measurements are associated with them, and they are terminated after 5 scans without measurements. The average length of a confirmed track which is eventually terminated is 22.3 scans. When excluding the tracks that are assumed to originate from actual targets, this is reduced to 15.7.

There is also an additional 47 tracks that die at the preliminary stage from the track initiation test. This brings the total number of confirmed and preliminary tracks to 65 for this scenario.

Track 3 is on Gunnerus. It remains stable on Gunnerus during the entire experiment. Tracks 6, 29, 37 and 45 originate from another target, which we refer to as the ToO-boat, for target of opportunity. This target was not equipped with an AIS transponder, and we have not been able to identify its callsign. We notice the open interval between tracks 6 and 29, which is due to ToO-boat passing through the radar shadow of Gunnerus. Another collection of tracks, 26, 44 and 65, originated from a buoy northwest of Gunnerus. The origin of track 17 is somewhat uncertain, but there is some evidence that it may be due to multi-path returns from Gunnerus, reflected by the mast or other installations on Telemetron.

Although 18 confirmed tracks may seem dramatic in a scenario where only two proper targets were present, it should be noted that most of these tracks originate from land, and stays close to it until they are terminated. This means that although the maps used to mask out detections from land is not perfect, they are sufficient for our purposes.

5.5 Clustering Radius Effect on Number of Measurements

The tracking results can be used to investigate how the assumption of maximum one measurement per target holds. This is done by considering the tracks of the two known targets in Figure 10. The number of measurements inside the validation gates are counted, and Figure 11 shows the number of gated measurements for Gunnerus and the target of opportunity with different clustering radii. There is clearly a difference between the boats, with Gunnerus having a heavier tail of measurements. We believe this to be caused by Gunnerus having both a significantly larger area than the ToO-boat, and being closer on the radar. Both of these effects make Gunnerus occupy more radar cells than the ToO-boat, which can lead to a more fragmented target if the clustering does not capture all the effects. It can also be caused by more subtle radar effects, such as sidelobes from the radar.

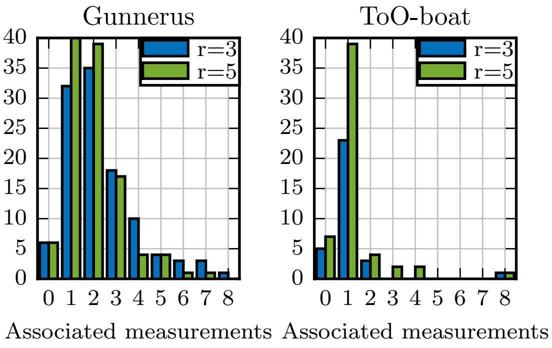

Fig. 11 Number of measurements in the validation gate for clustering radii 3 and 5 meters. Gunnerus on the left and the ToO-boat on the right.

6 Discussion and topics for future research

The results presented in the previous section lead to some implications and and suggestions for radar-based tracking in the context of maritime COLAV. We summarize these reflections in the four subsequent paragraphs.

Multi-target issues did not apear to be of critical importance in these experiments. Both in the "Trondheimsfjord II" scenario and the "Gunnerus" scenario, we never came across situations that involved confusion between tracks beyond what a parallel bank of PDAFs can handle.

Track continuity appears to be a more important issue. This is also a matter of data association, since deciding to keep or kill a track boils down to a decision about measurement origin hypotheses. The tracking community has a somewhat ambivalent attitude towards the importance of track continuity. While track continuity is a cornerstone of classical tracking methods such as PDAF, more recent methods based on random finite sets [10] do not provide continuous tracks. In any case, it could be argued that the track on ToO-boat should have been maintained for the 22 scans during which it resides in Gunnerus' radar shadow. However, standard existence-based tracking methods such as the IPDA would give the track extremely small existence probabilities under these circumstances, even if poor visibility was accounted for [3].

Target extent also appears to deserve closer scrutiny. The detrimental effect of radar shadows could be mitigated by modeling the radar shadow as part of the measurement model, by means of knowledge about the shadowing target's footprint. The radar footprint of an extended target may also carry other useful information that could aid both data association and state estimation.

The results demonstrate how false tracks inevitably occur in a tracking system. On the one hand, a COLAV system should be designed so that this has a minimal impact. On the other hand, more refined track quality measures can be useful to remove or discriminate false tracks. The track existence concepts of the IPDA [4] and random set methods [10] is a good starting point for this, and additional measures based on "track coherence" [3], consistency of target footprint, etc., may also be explored.

7 Conclusion

In this paper we have described all the building blocks of a complete PDAF-based radar tracking system for usage onboard an ASV. Experimental results demonstrate reasonable performance of the tracking system. While the tracking system described in this paper is mature enough to be used as part of experimental work on COLAV, further developments, especially with regard to track continuity, extended targets and track quality measures, should be conducted to make the tracking system truly reliable for COLAV.

Acknowledgements This work was supported by the Research Council of Norway through the projecs 223254 (Centre for Autonomous Marine Operations and Systems at NTNU) and the project 244116/O70 (Sensor Fusion and Collision Avoidance for Autonomous Marine Vehicles). The authors would like to express great gratitude to Kongsberg Maritime and Maritime Robotics for placing high-grade navigation technology and the Telemetron vehicle at our disposal, and especially Thomas Ingebretsen for help with implementing the software interfaces.

References

1. Y. Bar-Shalom and X.R. Li. *Multitarget-multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.
2. Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.
3. Edmund Brekke, Oddvar Hallingstad, and John Glattetre. The signal-to-noise ratio of human divers. In *Proceedings of OCEANS'10*, Sydney, Australia, May 2010.
4. Subhash Challa, Mark R. Morelande, Darko Mušicki, and Robin J. Evans. *Fundamentals of Object Tracking*. Cambridge University Press, 2011.
5. Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
6. Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.
7. P. P. Gandhi and S. A. Kassam. Analysis of CFAR processors in non-homogeneous background. *IEEE Transactions on Aerospace and Electronic Systems*, 24(4):427–445, July 1988.
8. Abbas Harati-Mokhtari, Alan Wall, Philip Brooks, and Jin Wang. Automatic identification system (AIS): Data reliability and human error implications. *Journal of Navigation*, 60(03):373, aug 2007.
9. X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking. Part I: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, oct 2003.
10. Ronald Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, MA, USA, 2007.
11. Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.
12. P.C. Niedfeldt and R.W. Beard. Multiple target tracking using recursive RANSAC. In *American Control Conference (ACC), 2014*, pages 3393–3398. Portland, OR, USA, June 2014.
13. G.W. Pulford. Taxonomy of multiple target tracking methods. *IEE Proceedings - Radar, Sonar and Navigation*, 152(5):291–304, 2005.
14. Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*. Kobe, Japan, 2009.
15. Petr Svec, Atul Thakur, Eric Raboin, Brual C. Shah, and Satyandra K. Gupta. Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. *Autonomous Robots*, 36(4):383 – 405, 2014.
16. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.
17. Ba-Ngu Vo, Mahendra Mallick, Yaakov Bar-Shalom, Stefano Coraluppi, Richard Osborne, Ronald Mahler, and Ba-Tuong Vo. Multitarget tracking. In *Wiley Encyclopedia of Electrical and Electronics Engineering*. Wiley, 2015.
18. Michael T. Wolf, Christopher Assad, Yoshiaki Kuwata, Andrew Howard, Hrand Aghazarian, David Zhu, Thomas Lu, Ashitey Trebi-Ollennu, and Terry Huntsberger. 360-degree visual detection and target tracking on an autonomous surface vehicle. *Journal of Field Robotics*, 27(6):819 – 833, 2010.

View publication stats