Multi-target tracking within large geographical areas – algorithms for improved accuracy and speed

Brita H. Hafskjold Gade
Norwedian Defence Research Establishment
Kjeller, Norway
brita.gade@ffi.no

Carina N. Vooren
Norwedian Defence Research Establishment
Kjeller, Norway
carina-norberg.vooren@ffi.no

Alexander Meyer Sjøberg
Norwedian Defence Research Establishment
Kjeller, Norway
alexander-meyer.sjoberg@ffi.no

Morten Kloster
Norwedian Defence Research Establishment
Kjeller, Norway
morten.kloster@ffi.no

***Abstract*—This paper addresses two challenges which are specific for tracking large amounts of vehicles within wide geographical areas. The first problem is related to the Earth's curvature, which may degrade the prediction accuracy over long distances. The second problem is concerning the gating process, which may be slow. This paper proposes a single target tracking algorithm that predicts along the great circle, and thus minimizes the unmodelled prediction error, and a multi target gating algorithm that speeds up the gating process by using a combination of clustering and the *n*-vector position representation.**

Keywords—Earth curvature, multi-target tracking, gating, clustering

I. INTRODUCTION

Merely a few decades ago, maritime surveillance was dependent upon land-based sensors, giving an acceptable maritime surveillance only relatively close to shore. In the last few decades, however, there has been a steep increase in the number of satellite-mounted sensors and information collection devices [1]. Examples are AIS (Automatic Identification System) [2][3][4] receivers, SAR (Synthetic Aperture Radar) [5], NRD (Navigation Radar Detector) [6] [7] and EO (Electro Optical) [8] sensors. These can be combined with land-based sensors for maritime surveillance in the vicinity of land [9][10][11], or used alone for global maritime surveillance [12][13].

The satellite mounted information sources (together with detection algorithms for the imaging sensors) give position information for a large number of marine vessels within wide geographical areas. Two of the challenges that are specific for tracking large amounts of vehicles within wide geographical areas are:

- *Earth curvature:* Due to the curvature of the Earth, the vehicles do not move along a straight line (as is prescribed in regular algorithms), even when having constant velocity and speed.
- *Gating:* The gating process (i.e., the process of matching a position measurement with the track(s) to which it may belong) may be slow due to the large number of tracks.

This paper suggests algorithms addressing these challenges. The algorithms are intended for tracking of (larger) marine vessels but can also be used for tracking other types of vessels if the vertical speed of the vessel is negligible when compared to the horizontal speed of the vessel.

A. Earth curvature

There are several alternatives for position representation, but they are not equally suited for multi-target tracking within large geographical areas.

1) NED and ENU

Many target-tracking applications use a flat-Earth approximation, where a single Cartesian coordinate frame is centred at a specific geographical position. Two popular alternatives are the ENU (East-North-Up) and the NED (North-East-Down) right-hand coordinate frames, where the north-east plane is a tangent plane to the Earth ellipsoid. Cartesian coordinate frames such as NED and ENU can be used directly in the state vector of the estimator part of the tracker. They work well for modelling of vehicle dynamics, since both velocity and acceleration vectors are easy to express, and since the horizontal and vertical directions are separated. While being a good solution for relatively small areas, local flat-Earth approximation may lead to growing errors as distances increase from the coordinate frame origin. Horizontal and vertical directions also deviate more from the correct directions as the distance from the origin increases.

2) Longitude and latitude

Representing positions by longitude, latitude and height makes it easy to separate horizontal and vertical directions, but they are seldom used in the state vector as there is no easy way to model the dynamics of a vehicle by use of longitude and latitude. This solution has singularities at the North and South poles and thus do not have a one-to-one property. In addition, calculations near ±180° longitude become complicated due to the discontinuity.

3) UTM/UPS

Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS) are map projections (i.e., a mapping of points on a curved surface to a plane) valid in a limited geographical area. These systems are well-defined, and their coordinate values approximately correspond to metres. However, due to the projection they have an inherent distortion and thus a corresponding error in many calculations. When crossing zones, general calculations also get very complex. Thus, UTM and UPS is not the preferred position representation to use in the state vector for tracking over large geographical areas.

4) Earth-centered Cartesian coordinate frames

Earth-centred Cartesian coordinate frames, such as the ECEF (Earth-Centred Earth-Fixed) coordinate frame is a popular global position representation for tracking, as the coordinates can be used directly in the state vector. ECEF gives a Cartesian 3D position vector that is non-singular and intuitive. However, the horizontal and vertical directions are not easily separated, which makes it more difficult to model vehicle dynamics, as most vehicles have different dynamics in the horizontal versus the vertical direction.

5) n-vector

n-vector [14][15] is an alternative global position representation, where the *n*-vector representation of a position B at the Earth's surface is defined as the outward pointing normal vector of the surface at B, with unit length. The surface here is the reference ellipsoid, such as WGS 84. As with latitude and longitude, the *n*-vector representation makes it easy to separate horizontal and vertical directions, but there is as yet no published easy way to model the dynamics of a vehicle by using *n*-vector. The *n*-vector is a one-to-one representation, i.e., any unit vector corresponds to one unique surface position, and any surface position corresponds to one unique *n*-vector. This one-to-one property is not held by latitude/longitude, which is a many-to-one representation at the poles. *n*-vector and height/depth give separate horizontal and vertical positions and have no singularities or discontinuities. In addition, *n*-vector makes it easy to find all three axes of the local Cartesian coordinate frame.

We have not found any literature presenting algorithms that use n-vector for target tracking or gating, even though our use of *n*-vector in target tracking was mentioned in [15].

B. Gating

For target tracking of large areas, where there may be thousands of targets, gating, i.e., the process of deciding whether a new measurement may belong to a track (or an earlier measurement) or not, is an important task. This task may be time-consuming since every new measurement needs to be gated against all existing tracks and earlier measurements. In section III, we propose a faster algorithm for the gating process.

C. Notation

For all vectors in this paper, we use an arrow above the vector to denote the coordinate-free form, and the following superscripts to denote a coordinate frame in which the vector is decomposed:

- *N*: the local NED frame
- *E*: an Earth-centred Earth-fixed coordinate frame, such as ECEF.

This paper shows equations with the use of NED as the local coordinate frame. The use of ENU would give similar equations.

D. Paper overview

This paper is organized as follows: We suggest a single-target-tracking algorithm that follows the Earth curvature; this algorithm is presented in Section II. To speed up the gating process for applications with many tracks in a wide geographical area, we propose a gating algorithm based on the use of clustering in section III. Section IV contains results for both the tracking algorithm and the gating algorithm. Section V concludes the paper.

II. TARGET TRACKING ALONG THE EARTH CURVATURE

In this section, we assume the tracking of a single target, where all associations are assumed to be correct.

Regular CV (Constant Velocity) predictors, whether they use ECEF or a local Cartesian coordinate frame, predict the future position of the vehicle along a straight line given by the direction of the estimated velocity vector. This is illustrated in Fig. 1, with a zoomed-in version in Fig. 2. In these figures, $p_{k|k}$ is the updated position at timestep k and $p_{k+1|k}$ is the predicted position. As can be seen, the curvature of the Earth gives an un-modelled prediction error. The measurement at time k, z_{k+1} lies on the Earth surface (here assuming a height equal to zero). The un-modelled prediction error can result in an error also in the updated position, $p_{k+1|k+1}$. By instead predicting along the great circle, the predicted position $p'_{k+1|k}$ will remain at the surface of the Earth, and so will the updated position $p'_{k+1|k+1}$.

For applications where the targets of interest are assumed to follow regular seaway routes, one could use prediction models as in [16]. In this paper, however, we do not make this assumption.

This section describes a tracking algorithm, which we call nGCP (*n*-vector Great Circle Prediction), using prediction along the great circle. Fig. 3 shows a flowchart of the algorithm, and each step of the algorithm is described in the following subsections.

A. Measurement

We assume that the measurement of the vehicle is given as a global position, as height (if applicable) and $\boldsymbol{n}^E_{k,meas}$, i.e., the *n*-vector of the global position at timestep k, decomposed in coordinate frame E. The measurement error is typically given as $\delta\boldsymbol{p}^N_{k,meas}$, i.e., decomposed in the local coordinate system N with the origin close to or at the position of the measurement, where $\delta\boldsymbol{p}^N_{k,meas} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}^N_k)$ and $\boldsymbol{R}^N_k$ is the covariance matrix.

B. Prediction (vehicle model)

Our predictor assumes constant speed, similarly to the standard CV model. It predicts along the great circle in the same angular direction as the velocity vector. The following equation for the predicted position, i.e., the a priori position (at time k+1), is based on example 8 in [17] but changed to also work for positions at the poles.

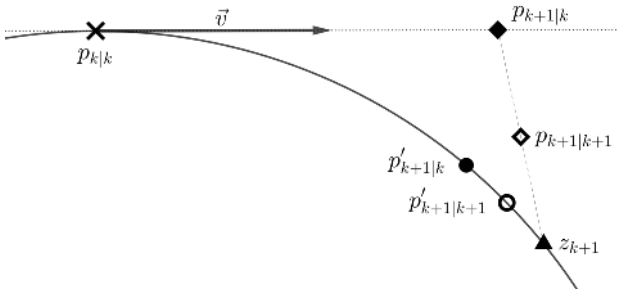


Fig. 1. Illustration: The drawback of predicting along the direction of the velocity vector.

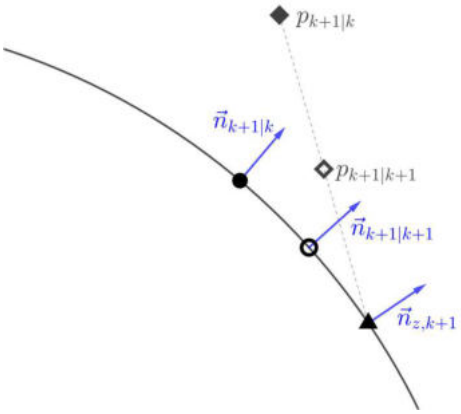


Fig. 2. Illustration: A zoomed-in version of Fig. 1, where the *n*-vector representation of the positions at the Earth surface is shown.

$$\boldsymbol{n}^E_{k+1|k} = \boldsymbol{n}^E_{k|k} \cdot \cos\left(\frac{|\boldsymbol{v}_{k|k}| \cdot \Delta t}{r + h_{k|k}}\right) + \text{unit}(\boldsymbol{v}^E_{k|k}) \cdot \sin\left(\frac{|\boldsymbol{v}_{k|k}| \cdot \Delta t}{r + h_{k|k}}\right) \tag{1}$$

In (1), 'unit()' makes the input vector a unit vector (of length 1), $\boldsymbol{n}^E_{k|k}$ and $h_{k|k}$ represents the horizontal and vertical parts of the a posteriori position at time k, r is the mean Earth radius, $\boldsymbol{v}^E_{k|k}$ is the a posteriori velocity vector decomposed in E, and $|\boldsymbol{v}_{k|k}|$ is the length of this a posteriori velocity (since we only need the length here, the coordinate system in which the velocity vector is decomposed is irrelevant). The a posteriori velocity vector at time k, decomposed in N, is given by:

$$\boldsymbol{v}^N_{k|k} = \boldsymbol{v}^N_{k|k-1} - \widehat{\delta \boldsymbol{v}}^{\boldsymbol{N}}_{k|k-1} \tag{2}$$

where $\widehat{\delta v}^{\boldsymbol{N}}_{k|k-1}$ is the output from the error state updater, and the local coordinate system N is centred in the predicted position $\boldsymbol{n}^E_{k+1|k}$.

C. Innovation calculation

Since the measurement $\boldsymbol{n}^E_{k,meas}$ is given as a position, the a priori position $\boldsymbol{n}^E_{k|k-1}$ from (1) is equal to the predicted measurement. The position difference between these (including their respective heights), decomposed in N, can easily be found by using Example 1 in [17]. The innovation given by this position difference is equal to the difference between the prediction error and the measurement error:

$$\delta \boldsymbol{p}^N_{k,meas} - \delta \boldsymbol{p}^N_{k|k-1} \tag{3}$$

Here $\delta \boldsymbol{p}^N_{k|k-1}$ is the error of the position represented by $\boldsymbol{n}^E_{k|k-1}$ in (1) (and the respective height), and $\delta \boldsymbol{p}^N_{k,meas}$ is the measurement error.

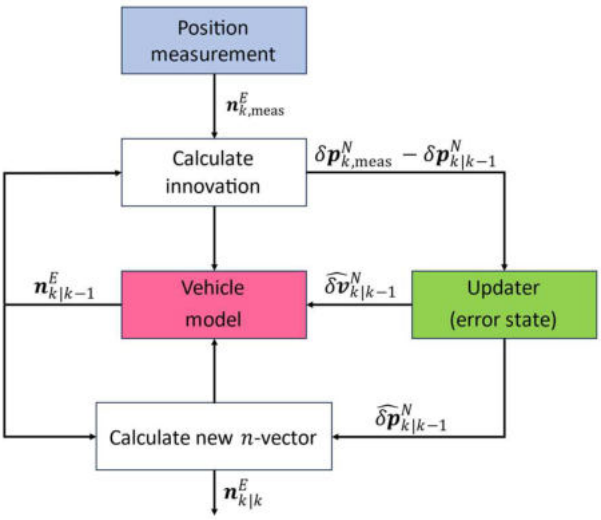


Fig. 3. Flowchart of the proposed nGCP algorithm

D. Updater

The updater estimates the unknown error $\widehat{\delta \boldsymbol{p}}^{\boldsymbol{N}}_{k|k-1}$ of the a priori position and the error $\widehat{\delta \boldsymbol{v}}^{\boldsymbol{N}}_{k|k-1}$ of the predicted velocity as

$$\begin{bmatrix} \widehat{\delta \boldsymbol{p}}^{\boldsymbol{N}}_{k|k-1} \\ \widehat{\delta \boldsymbol{v}}^{\boldsymbol{N}}_{k|k-1} \end{bmatrix} = \boldsymbol{K} \cdot \left(\delta \boldsymbol{p}^N_{k|k-1} - \delta \boldsymbol{p}^N_{k,meas}\right) \tag{4}$$

where the Kalman gain $\boldsymbol{K}$ is calculated by the standard Kalman filter equations, using $\boldsymbol{R}^N_k$ and the process noise covariance matrix as input.

The nGCP algorithm uses the flat-Earth approximation only for the updater, where the position differences are small, and the corresponding errors due to the Earth curvature are negligible.

E. Calculate new n-vector

The *n*-vector of the a posteriori position, $\boldsymbol{n}^E_{k|k}$, is calculated by using example 2 of [17], by using $\boldsymbol{n}^E_{k|k-1}$ and $\widehat{\delta \boldsymbol{p}}^{\boldsymbol{N}}_{k|k-1}$ as input.

III. Algorithm for Fast Gating within Large Geographical Areas

We propose a gating algorithm to speed up the gating process. This gating algorithm builds upon the traditional clustering method as described in [18]. The traditional clustering algorithm is illustrated by the flowchart in Fig. 4, and has the following steps:

- A new cluster is set up for the first measurement we receive.
- A new measurement is added to a cluster if it falls within the gate of an existing track or measurement in the cluster.
- A new cluster is set up for each measurement that is not added to an existing cluster.

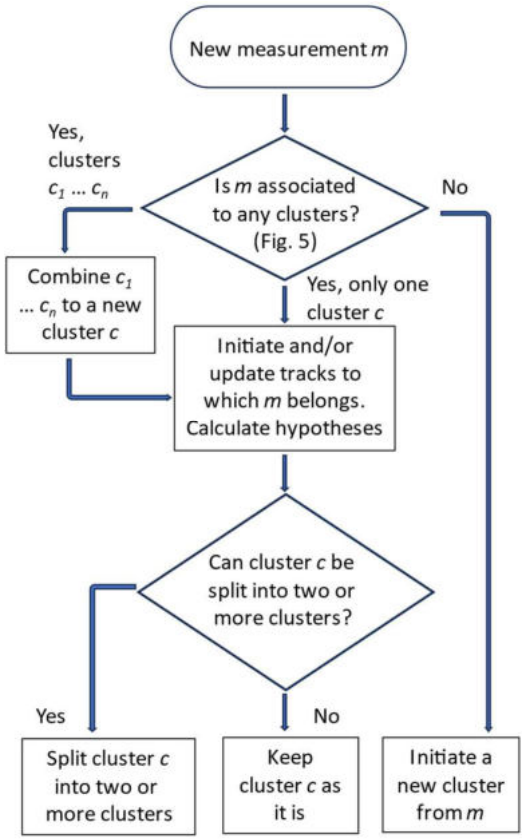


Fig. 4. Traditional clustering as described in [18]

- If a measurement is added to more than one cluster, these clusters are combined to a new, larger cluster.
- A cluster may also be divided into smaller clusters. For HOMHT (Hypothesis Oriented MHT), if a track is contained in all hypotheses in a cluster, this track leaves the cluster, and a new cluster is set up for this track.

The proposed gating process is described in subsection III.*A*. To speed up the gating process, we propose to use a border for each cluster, as described in subsections III.*B* and III.*C*. We also propose methods to speed up the gating process for measurement-to-measurement gating and measurement-to-track gating, as described in subsections III.*D* and III.*E*, respectively. If the new measurement is not outside the gates in section III.*D* and III.*E*, we calculate the Mahalanobis distance as described in subsection III.*F*.

A. Gating algorithm

Regular gating processes calculate the Mahalanobis distance for all possible associations, which can be time-consuming if there is a large number of measurements and tracks. Gating in stages, where larger gates are used before smaller gates, are widely used, but we have not found earlier work where it is used in combination with clusters.

The aim of our gating algorithm is to speed up the calculations, mostly by calculating the Mahalanobis distance as few times as possible.

The main steps of the gating algorithm are shown in the flow chart in Fig. 5. First, the cluster border is used to make a fast decision of whether measurement *m* may belong to the cluster. For the clusters in which the measurement is inside

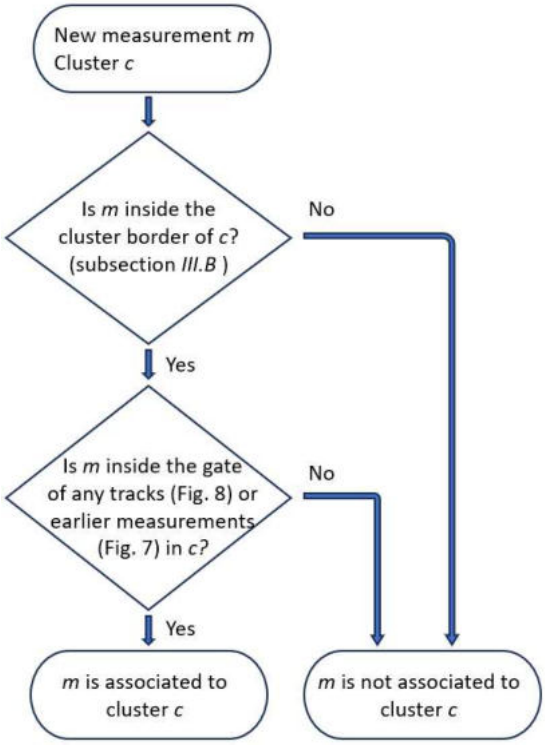


Fig. 5. The main steps of the proposed gating algorithm

the border, the methods in sections *III.D* and *III.E* are used to make fast decisions as to which gates of tracks or earlier measurements that *m* lies within.

B. Cluster border using n*-vector*

We use cluster borders to speed up the gating process. Since we have not found any literature on cluster borders, we here define our cluster border as a geographical circle on the Earth's surface, as illustrated in Fig 6. The circle is large enough to cover all the uncertainty areas of all tracks and earlier measurements in the cluster. In this way, a measurement that falls outside the cluster border clearly does not belong to the cluster, but for a measurement that falls within the cluster border we still need to perform measurement-to-measurement-gating and/or measurement-to-track gating to decide whether it belongs to the cluster.

We define the placement and size of the cluster border of cluster c by $\vec{n}_{c,\,\text{center}}$, which is an n-vector for its center position, and an angle β_c corresponding to the angle between $\vec{n}_{c,\,\text{center}}$ and an arbitrary n-vector on the border circle, $\vec{n}_{c,\,\text{border}}$, as illustrated in Fig. 6. Since n-vectors have length 1, the angle β_c is thus defined by

$$\beta_c = \cos^{-1}\left(\boldsymbol{n}^E_{c,\text{center}} \cdot \boldsymbol{n}^E_{c,\text{border}}\right) \qquad (5)$$

$$= \sin^{-1}\left(\left|\boldsymbol{n}^E_{c,\text{center}} \times \boldsymbol{n}^E_{c,\text{border}}\right|\right)$$

where the latter formula is more accurate for small angles.

The vector $\vec{n}_{c,\,\text{center}}$ decomposed in E, $\boldsymbol{n}^E_{c,\text{center}}$, is calculated as the horizontal geographical midpoint of the n-vectors of all tracks and earlier measurements belonging to the cluster (equation based on example 7 in [17])

$$\boldsymbol{n}^E_{c,\,\text{center}} = \text{unit}\left(\sum_{i=1}^{m} \boldsymbol{n}^E_i\right) \qquad (6)$$

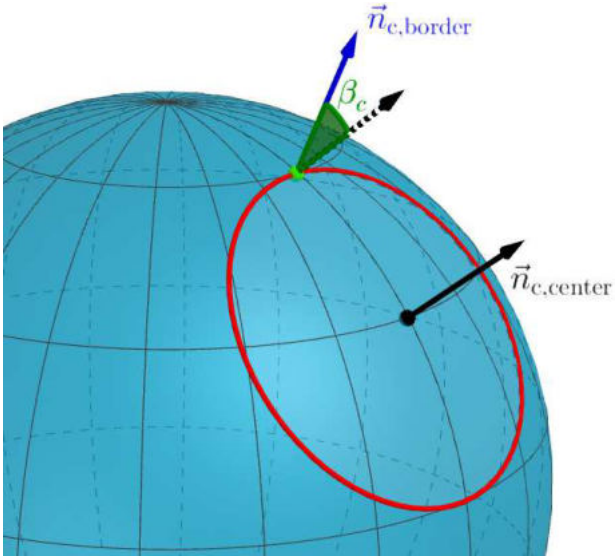


Fig. 6. The center of the cluster is given by an n-vector, while the size of the border circle is given by the angle between the center n-vector and an arbitrary n-vector on the border circle.

where $\boldsymbol{n}_i^E$ is the position of track or measurement number *i*.

The angle β_c is found by

$$\beta_c = \max_b\left(\cos^{-1}\left(\boldsymbol{n}_{c,\text{center}}^E \cdot \boldsymbol{n}_{c,b}^E\right) + q_b\right) \tag{7}$$

where $\boldsymbol{n}_{c,b}^E$ is the position (either predicted or updated) of track b at time t_b and

$$q_b = \left(l_{\max,b} + v_{\max} \cdot (t - t_b)\right)/r \tag{8}$$

where $v_{\max}$ is the assumed maximum velocity of the vehicle, *t* is the time of the new cluster border and $l_{\max,b}$ is the longest distance from track b to a point on its uncertainty ellipse, where the size of the uncertainty ellipse is determined by the chosen gate size, which is the maximal allowed statistical distance between a track and a new measurement. Since prediction of all tracks in the cluster may be time-consuming, we use the term $v_{\max} \cdot (t - t_b)$ in (8) to avoid having to predict all tracks in a cluster each time we compute a new cluster border.

C. *Is measurement* m *inside the cluster border of cluster* c*?*

To decide whether measurement *m* is inside the cluster borders of cluster *c*, we first calculate the variable *qm* as:

$$q_m = l_{\max,m}/r \tag{9}$$

where *r* is the mean Earth radius and $l_{\max,m}$ is the longest distance from *m* to a point on its uncertainty ellipse, where the size of the uncertainty ellipse is determined by the chosen gate size, which is the maximal allowed statistical distance. A new measurement is thus outside of the cluster border if

$$\gamma_{m,c} > q_m + \beta_c \tag{10}$$

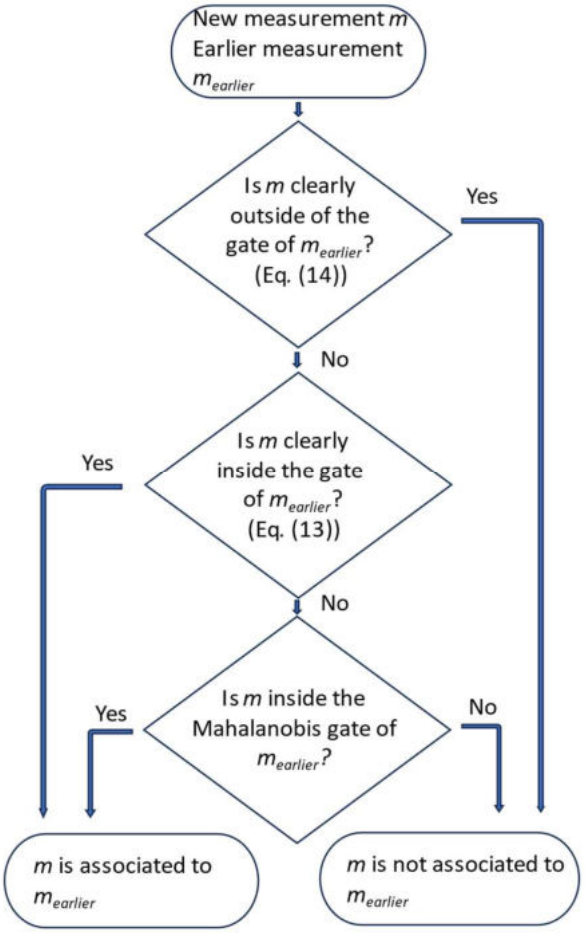


Fig. 7. A faster method for determining whether measurement *m* is inside the gate of earlier measurement $m_{earlier}$

where $\gamma_{m,c}$ is the angle between $\boldsymbol{n}_{c,center}^E$ and $\boldsymbol{n}_{meas}^E$, calculated in the same way as in (5).

D. *Is* m *inside the gate of earlier measurement* m_earlier*?*

To speed up the process of finding the earlier measurement(s) that the new measurement *m* is associated to, we first divide the associations into three categories, as shown by the flowchart in Fig. 7:

- Associations that are clearly inside the gate
- Associations that are clearly outside of the gate
- All other associations where we need to calculate the Mahalanobis distance as described in subsection III.*F*.

To decide whether *m* and the earlier measurement $m_{earlier}$ can clearly be associated, we first calculate the maximal distance that the vehicle may move during $\Delta t = t_m - t_{m\_earlier}$:

$$s = \Delta t \cdot v_{\max} \tag{11}$$

The horizontal position of the two measurements *m* and *m_earlier* are given by *n*-vectors decomposed in *E* as $\boldsymbol{n}_m^E$ and $\boldsymbol{n}_{m\_earlier}^E$, respectively. The Euclidean distance – assuming spherical Earth – between these positions is (by using Example 5 in [17])

$$s_{Euclidean} = \left|\boldsymbol{n}_m^E - \boldsymbol{n}_{m\_earlier}^E\right| \cdot r \quad (12)$$

Thus, *m* is clearly inside the gate of *m_earlier* if

$$s/r > \left|\boldsymbol{n}_m^E - \boldsymbol{n}_{m\_earlier}^E\right| \quad (13)$$

Note that for (11), *s* should be small compared to *r*. We also have that *m* is clearly outside of the gate of *m_earlier* if

$$s/r + q_{measA} + q_{measB} < \left|\boldsymbol{n}_{measB}^E - \boldsymbol{n}_{measA}^E\right| \quad (14)$$

where q_{measA} and q_{measB} can be found by using (7).

E. *Does* m *belong to track* b*?*

To speed up the process of finding the tracks that the new measurement *m* is associated to, we first divide the associations into two categories, as shown by the flowchart in Fig. 8:

- Associations that are clearly inside the gate
- Other associations where we need to calculate the Mahalanobis distance as described in subsection *III.F*.

In the same way as for (14), *m* is clearly outside the gate of track *b* if:

$$q_b + q_m < \left|\boldsymbol{n}_m^E - \boldsymbol{n}_b^E\right| \quad (15)$$

where q_b and q_m are found analogously to (9) for track *b* and measurement *m*, and $\boldsymbol{n}_b^E$ is the horizontal position of track *b*.

F. *Gating by using Mahalanobis*

For all tracks where the gating cannot be determined by (13), (14) or (15), we need to calculate the Mahalanobis distance *d* between *m* and *b*. Since this calculation is only performed for points that are geographically close, we can use a local flat-Earth approximation, such that the Mahalanobis distance can be found by

$$d^2 = \left(\boldsymbol{p}_{b,m}^N\right)^T \left(\boldsymbol{C}_b^N + \boldsymbol{R}_m^N\right)^{-1} \left(\boldsymbol{p}_{b,m}^N\right) \quad (16)$$

where $\boldsymbol{C}_b^N$ is the covariance matrix of the error of the predicted position of track *b*, decomposed in a NED coordinate system with origin in *b*, $\boldsymbol{R}_m^N$ is the measurement noise covariance matrix decomposed in a NED coordinate system with origin in *m*, and the position vector $\boldsymbol{p}_{b,m}^N$ from *b* to *m* is calculated in the same way as for calculating the innovations in section II.*B*. We can assume the rotation difference of the NED coordinate systems of *b* and *m* to be small enough that we may sum $\boldsymbol{C}_b^N$ and $\boldsymbol{C}_m^N$ as if they were in the same NED coordinate system. The error from this assumption will be much lower than the uncertainty of the covariance matrices.

IV. Results

A. *Accuracy*

If a vehicle is predicted a distance of 250 km (i.e., less than 1/10 of the Norwegian Exclusive Economic Zone (EEZ) in the north/east direction), the unmodelled prediction error, if predicting along the direction of the estimated velocity vector, will be approximately 5 km, as shown in Fig. 9.

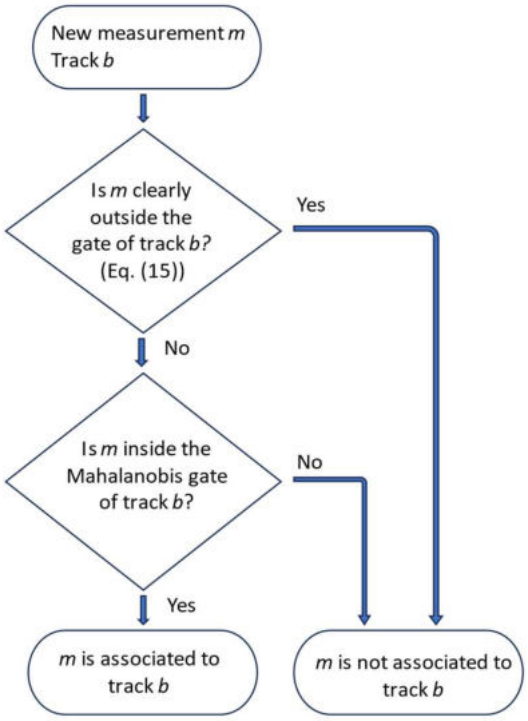


Fig. 8. A faster method for determining whether measurement *m* is inside the gate of track *b*.

Also for shorter distances, the non-modelled prediction error may need to be taken into consideration, as shown in Fig. 10. The nGCP algorithm avoids this prediction error by predicting along the great circle. Since the Earth is not a perfect sphere, there is also a small difference between a great circle distance and the distance along the geodesic on the spheroid (e.g., WGS 84) describing the Earth. Since the calculation of the distance along the spheroid requires iterative formulas, this difference is not taken into consideration here.

B. *Computational burden*

This subsection presents the computation time for an implementation in Matlab when run on a Intel(R) Core(TM) i5-8265U CPU. We have simulated the prediction, the gating, and update steps for both the regular gating process, using

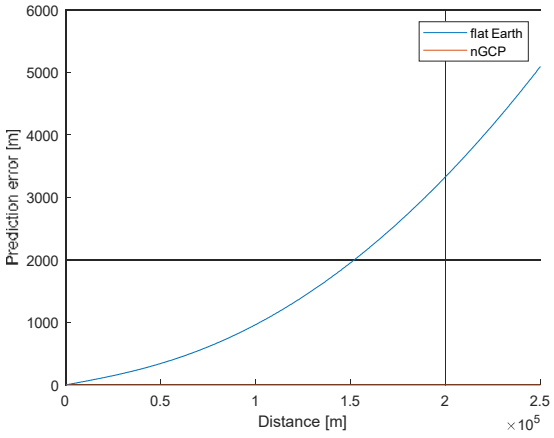


Fig. 9. Prediction error for long distances, when predicting along the velocity vector direction, for a velocity vector with zero height component.

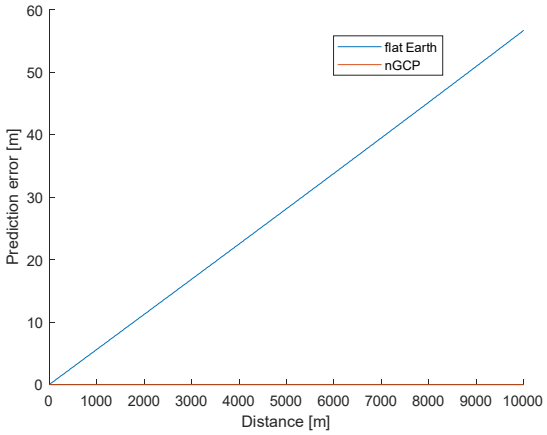


Fig. 10. Prediction error for medium distances, when predicting along the velocity vector direction, for a velocity vector with zero height component.

ECEF coordinates and a standard CV Kalman Filter, and the proposed nGCP algorithm with the proposed gating process.

The computation time for running the prediction step 200 times is 0.0145 seconds for the nGCP algorithm and 0.0158 seconds for the regular prediction algorithm, as an average of 1000 runs.

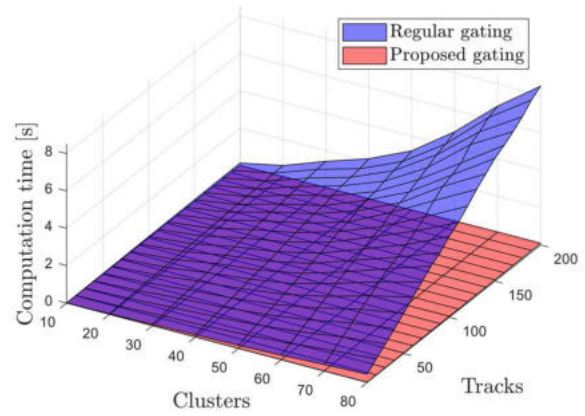
The computation time for running the update step 200 times is 0.0088 seconds for the nGCP algorithm and 0.000022 seconds for the regular update algorithm, as an average of 1000 runs.

We have simulated the gating and update process for a case with n_c clusters, each containing n_t tracks, where we receive n_c measurements. Fig. 11 shows the computation time for n_c ranging from 10 to 80 and n_t ranging from 1 to 200. In Fig. 11 (a), the regular gating calculates the Mahalanobis distance for every track-measurement pair, and in Fig. 11 (b), the regular gating is a two-stage process. This two-stage process consists of a cubic gate (in the x-, y- and z-coordinates of the ECEF position) that is large enough to ensure that all measurements fall within the cubic gate of the correct track, and the calculation of the Mahalanobis distance for measurements falling within this cube. Even though the update step is slower for the nGCP algorithm than for the regular algorithm, this has little impact on the combined computation time of both the gating and the update algorithms, as can be seen from Fig. 11.

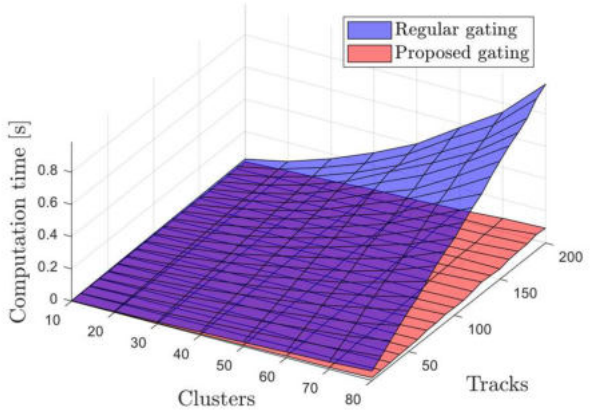
In Fig 11 (c), the proposed gating is compared to a two-stage process that consists of a gate in latitude and longitude (which is rectangular for areas near the equator, and more triangle-shaped near the poles), and the calculation of the Mahalanobis distance for measurements falling within this first gate. This gating process is slower than the other gating methods since it requires that the position of all tracks are converted from ECEF to latitude and longitude.

V. CONCLUSION:

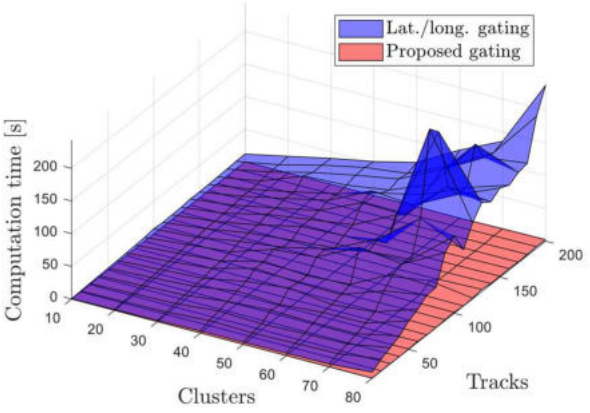
A single target tracking algorithm that predicts along the great circle is proposed. The algorithm avoids the unmodelled prediction error that follows from predicting along the direction of the velocity vector.



(a) Computation time for the proposed gating vs regular gating, where the regular gating calculates the Mahalanobis distance for every track-measurement pair.



(b) Computation time for the proposed gating vs regular gating, where the regular gating is a two-stage process, using a first gate in ECEF.



(c) Computation time for the proposed gating vs a two-stage gating with a first gate in longitude and latitude.

Fig. 11. Computation time for the combination of the gating process and the update step, for a range of number of clusters and a range of number of tracks per cluster.

A gating algorithm is proposed for tracking within large geographical areas with a large number of vehicles. This gating algorithm has shown a considerable speed-up when compared to traditional gating algorithms, especially for large numbers of tracks and clusters.

VI. REFERENCES

[1] G. Soldi et al., "Space-Based Global Maritime Surveillance. Part I: Satellite Technologies," in IEEE Aerospace and Electronic Systems Magazine, vol. 36, no. 9, pp. 8-28, 1 Sept. 2021, doi: 10.1109/MAES.2021.3070862.

[2] International Maritime Organization, "AIS transponders", https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx [Online; accessed feb 14, 2023]

[3] Skauen, Andreas Nordmo, "Quantifying the tracking capability of space-based AIS systems", Advances in Space Research; Volume 57.(2), pp 527-542, 2016.

[4] Eriksen, Torkild; Greidanus, Harm; Delaney, Conor, "Metrics and provider-based results for completeness and temporal resolution of satellite-based AIS services", Marine Policy, Volume 93, pp 80-92, 2018.

[5] Arsenios Tsokas, Maciej Rysz, Panos M. Pardalos, and Kathleen Dipple. SAR data applications in earth observation: An overview. *Expert Systems with Applications*, 205:117342, 2022.

[6] Norwegian Defence Research Establishment (FFI) "Norsat-3, ship surveillance with a navigation radar detector", *Facts*, 2021.

[7] TNO (Dutch Organization for Applied Scientific Research), "MilSpace2: Spectrum monitoring dual satellite system", https://www.tno.nl/en/newsroom/2022/12/milspace2-spectrum-monitoring-dual, 2022, [Online; accessed Mar 01, 2024].

[8] Madry, S., Pelton, J.N., "Electro-optical and Hyper-spectral Remote Sensing", in: Pelton, J.N., Madry, S., Camacho-Lara, S. (eds) Handbook of Satellite Applications, Springer, New York, NY, 2013.

[9] M. Guerriero, P. Willett, S. Coraluppi and C. Carthel, "Radar/AIS data fusion and SAR tasking for Maritime Surveillance," 2008 11th International Conference on Information Fusion, Cologne, Germany, pp. 1-5, 2008

[10] D. Gaglione, P. Braca and G. Soldi, "Belief Propagation Based AIS/Radar Data Fusion for Multi - Target Tracking," 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 2018, pp. 2143-2150, doi: 10.23919/ICIF.2018.8455217.

[11] Gaglione, D., Soldi, G., Meyer, F., Hlawatsch, F., Braca, P., Farina, A. and Win, M.Z. (2020), Bayesian information fusion and multitarget tracking for maritime situational awareness. IET Radar Sonar Navig., 14: 1845-1857.

[12] G. Soldi et al., "Space-Based Global Maritime Surveillance. Part II: Artificial Intelligence and Data Fusion Techniques," in IEEE Aerospace and Electronic Systems Magazine, vol. 36, no. 9, pp. 30-42, 1 Sept. 2021, doi: 10.1109/MAES.2021.3070884.

[13] A. Milios, K. Bereta, K. Chatzikokolakis, D. Zissis and S. Matwin, "Automatic Fusion of Satellite Imagery and AIS data for Vessel Detection," 2019 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, pp. 1-5, 2019.

[14] B. H. H. Gade and K. Gade, "n-vector – formulas with derivations", FFI/RAPPORT 2007/00633, Norwegian Defence Research Establishment (FFI).

[15] K. Gade," A Non-singular Horizontal Position Representation", The Journal of Navigation, Volume 63, Issue 03, pp 395-417, July 2010.

[16] L. M. Millefiori, G. Pallotta, P. Braca, S. Horn and K. Bryan, "Validation of the Ornstein-Uhlenbeck route propagation model in the Mediterranean Sea," OCEANS 2015 - Genova, Genova, Italy, 2015, pp. 1-6, doi: 10.1109/OCEANS-Genova.2015.7271565.

[17] Norwegian Defence Research Establishment (FFI), "The n-vector page", [Online]. Available: https://www.ffi.no/en/research/n-vector.

[18] D. Reid, "An algorithm for tracking multiple targets," IEEE Transactions on Automatic Control, vol. 24, no. 6, pp. 843-854, December 1979, doi: 10.1109/TAC.1979.1102177.