



A new approach to map-assisted Bayesian tracking filtering

Jaime López-Araquistain*, Ángel J. Jarama, Juan A. Besada, Gonzalo de Miguel, José R. Casar

SSR-GPDS, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Keywords:

MHT
Ground target tracking
IMM
Non-linear tracking
Context based tracking

ABSTRACT

This paper presents a new non-linear filter designed to track targets following a road network, taking advantage of the road map information. The algorithm is based on a Bayesian Multiple Hypotheses modelling of movement process, postulating and evaluating different hypotheses on the segments being followed by the target after road junctions. Then, the along-road tracking is carried out, for each hypothesis, by a longitudinal IMM filter capable of tracking target movements along straight roads, circular segments, and generic curvilinear segments defined through Bézier curves. The algorithm also includes a lateral drift estimator, which tracks the lateral motion of the target with respect to road axis, to be able to estimate target piloting error and especially to track targets in wide roads. The paper completely describes the filter and associated measurement preprocessing procedures, and also includes a comparative evaluation of the proposed filter with other filtering methods in the literature.

1. Introduction

Tracking of targets following a road network on a 2D surface (i.e. ground) is a classical data-fusion application [1–3]. The use of road geometrical information as contextual data enabling an improvement of the tracking performance has shown its effectivity in practical cases in the literature [4,5]. Applications of filters using this kind of constraint, among others, are tracking of aircraft taxiing in an airport surface [6], military surveillance of ground targets following a road network [7,8] or indoor mobile tracking of users moving along corridors [9].

Many different methods for incorporating map constraints to target tracking have, therefore, been explored in the literature. Among them, we could define two distinct approaches:

1. Maintaining a full-state vector filtering, tracking at the same time along-road and transversal to road movement.
2. Decoupling the movement tracking in two components, one associated to along-road movement, and another one related to movement transversal to road (including here the capability to distinguish the road segment where the target lies).

Regarding the first approach, some potential ways to integrate road information in the full tracking filter would be: including heading/track angle pseudo-measures aligned with the road [10–12]; forcing the transition model to follow the track angle [6,13]; modifying the velocity-related state vector components to fulfil the road associated heading constraint [14]; including non-linearities making the state

vector compatible with both the position and the heading constraints imposed by the road [4,15–19]; or making input measurements compatible with the road network by projecting them in suitable road segments [6]. In all those approaches, the state vector used for 2D tracking has two positional components, two velocity components and (quite often) two acceleration components. Depending on the application and on the filter mode they can be expressed in different coordinate frames or using the most suitable mathematical representation (i.e. position in Cartesian coordinates, velocity in Cartesian or polar coordinates, acceleration in inertial Cartesian or along-track plus cross-track components, or converted to longitudinal acceleration and turn rate, etc.). Of course, in multiple model filters making use of this approach some of the model tuned filters could have reduced state vectors, without all the actual components of the 2D movement (e.g. constant velocity models without accelerations, stop models without velocity and acceleration, etc.).

The second approach (decoupled longitudinal and transversal tracking) is based on making a definition of the movement process compatible with the road network by design. It is based on the observation that in a road network the usual target movement is not longer an unconstrained 2D movement, but can be better understood as a 1D movement along the road, with minor transversal piloting corrections. Those corrections, made by the pilot to keep the target in its lane or in the road define the lateral movement of the target (to be called lateral drift in this paper) when it is moving along a road, making the target trajectory almost parallel to the road axis. An example in the literature of this kind of approach would be [20], where a filter somehow

* Corresponding author.

E-mail address: jaime.lopez@grpss.ssr.upm.es (J. López-Araquistain).

similar to our proposal, based on a variable structure IMM (VS-IMM) is described. Our map definition is more general since the one described before was not able to model generic curves. Therefore new measurement projection processes, based on a maximum likelihood approach, were designed. The filter uses a road map description based on three different kinds of road segments: straight, circle arcs, and Bézier curves.

An additional problem to be analysed in both approaches is how the proposed filters should work when the target approaches road junctions. Additional uncertainty arises from the fact the output road to be followed by the target is not known by the filter. Therefore we may make use of two strategies to address this problem: either disabling the non-linearities described previously until the output road segment is clear [6], or branching the filter into different filters adapted to each of the alternative road segments leaving the junction [15]. The first strategy is especially appropriate for full-state filtering approaches, while the second is suited for decoupled longitudinal and transversal tracking. The Multiple Hypothesis Tracking (MHT) procedure has typically used for association [21–23] but in this paper we proposed to use a MHT inspired procedure for following the branching at junctions, to be called Map-Based Multiple Hypothesis Filtering (MBMHF). The description in the paper fully covers all the hypotheses formation and evaluation processes, including new terms in the evaluation process resulting in improved performance in junctions vicinity, and making it different to the one described in [20].

Also, our method tracks the lateral drift, which is neglected in the described work (and in the rest of the relevant literature).

Summarising, the proposed filter follows the decoupled longitudinal and transversal tracking approach, implementing branching at junctions through an MBMHF approach, along-segment tracking through IMM filters, lateral drift-tracking through a Kalman filter, and measurement projections into generic curve road axes using a maximum likelihood approach.

The paper is organised as follows. In Section 2 the 2D tracking problem for targets following a road network is presented. Then, Section 3 explains the main components of the tracking system at a high level, and specially how they interact. After that, we follow a bottom-up approach to detail those components. First, the coordinates transformation needed to project the sensor measures in a road segment and to project tracks following the segments back into 2D are explained in Section 4. Afterwards, Section 5 presents the along-segment IMM filter, valid for a road without junctions. To continue with the description of the algorithms Section 6 describes, based on previous explanations, the projection processes in the road network and the MBMHF based branching. Later, the transversal offset tracker, and the associated coordinate transformations are explained in Section 7. In Section 8, some simulation and real data results are discussed, comparing the performance with other tracking systems. Finally, Section 9 summarises the main conclusions of this work.

2. Map-Based tracking problem statement

In this section we will describe the main interface and functional requirements of a multisensor tracking system following targets in 2D making use of map of road networks constraining the tracked objects movements. To do so, next we will first detail system inputs and outputs, and road network map modelling.

The system has 2D position measures as inputs. It is assumed that all the available sensors provide full 2D information in Cartesian coordinates, or that those measures are transformed in the Cartesian plane (i.e. from polar measures in the case of radar sensors). It is also assumed in this paper that the measures are perfectly associated beforehand target, so that we are not dealing with the association process. Additionally, the measures from the different sensors are assumed to be unbiased. This last assumption is not very realistic on many applications, since different sensors have different measure centroids, but bias can be reduced to be negligible tuning bias estimation and correction

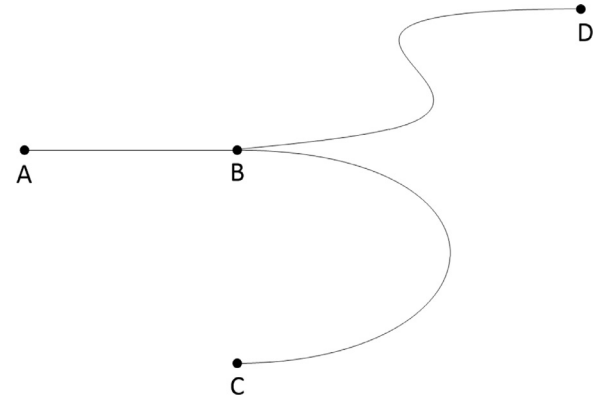


Fig. 1. Map example with the three basic elements.

methods such as those described in [30] to the specific application.

The system provides as output the identification of the segment of the road network where the target lies with highest probability (and the associated probability, as assessed by the system), and its position and velocity in Cartesian coordinates.

A key aspect of the system is the way to describe the kinematic constraints imposed by the presence of the road network. To describe those constraints we will start by describing the model of the network map. Any road-network map, whether it belongs to an airport or a highway network, or even a plan describing a set of corridors within a building, can be parameterised with simpler elements which allow the description of different parts of the map. In particular, in our approach, each road segment may be represented with three bidimensional basic elements: straight lines, circular arcs and splines [24]. These three kinds of segments are general enough to represent any map with precision. In Fig. 1 examples of those three elements are drawn. Specifically, the segment going from point A to point B is a straight line, the segment from point B to C is a circular arc, and the segment between B and D is defined through a spline. There are many options for the splines implementation. In this research we have selected the Bézier splines, and specifically cubic Bézier splines [25], due to its simplicity.

Every segment is delimited by an origin node and a destination node, being the algorithms to be described capable of performing tracking in both movement directions (from origin to destination node, or from destination to origin node). In fact, the system output is not affected by the decision of which end of each segment is designated as origin or destination.

Next we will describe the representation of each of the types of segments:

1. Straight lines: They are represented by an origin and a destination, defined in their Cartesian coordinates.
2. Circular arc: This element allows to represent simple map curves, through the origin point, the destination point, the centre of the curve and (implicitly) the radius.
3. Cubic Bézier Curves: They are used to define more complex map curves interpolating a collection segment samples, being defined in parametric form and smooth C2. Being smooth C2 implies that the derivatives f' and f'' exist and are continuous. A Bézier curve is formed by different consecutive sections, one between each pair of segment samples. In each section, s_i , the parametric form of the curve is as follows:

$$\begin{cases} x = a_{x1,i}\lambda^3 + a_{x2,i}\lambda^2 + a_{x3,i}\lambda + a_{x4,i} \\ y = a_{y1,i}\lambda^3 + a_{y2,i}\lambda^2 + a_{y3,i}\lambda + a_{y4,i} \end{cases} \quad \lambda \in [0, 1] \quad (1)$$

where $(a_{xj,i}, a_{yj,i})_{j=1...4}$ are the coefficients associated to the polynomial defining the coordinates (x, y) in the section s_i , while λ is the independent parameter allowing to traverse the section s_i as it

changes from 0 to 1. So, for $\lambda = 0$ we have the segment sample at the beginning of the section, and $\lambda = 1$ may be used to obtain the final segment sample of the section. The coefficients have to guarantee the smooth between sections, hence they are chosen carefully, following the procedures described in [24]. It should be noted that in order to define a point in this kind of segments both the relevant section and the λ parameter for the relevant section need to be identified.

In our conceptual model of the tracking problem we assume the map model to be complete (it models all relevant real roads, and no segment is unrelated to any real road). Also, the filter assumes the segment-based geometric model of the road is perfect. Note that due to the possibility to link collections of Cubic Bézier Curves, very reduced error is attainable.

3. Map-Based tracking filter overall structure

The filter decouples the tracking in two components, one related to along-road movement, and another one designed to track movement transversal to road. The along-road tracking process assumes that the target moves in parallel to the centre line of the road, focusing in tracking the projection of target position on this centre line. Therefore it preprocesses the measures by projecting them to this centre line. The transversal movement (perpendicular to this centre line) is estimated by a second part of the system, using as input the difference between the sensor measure and the projected measure.

The 2D position of the target may be represented by three values:

1. A segment identifier (s_{ID}), unambiguously defining the segment where the target position is being projected.
2. The distance along the segment, measured from the segment origin node, to be called d_{long} .
3. The signed transversal position (d_{trans} in the following) allows to completely define the target position. Positive d_{trans} means the offset is to the right of movement direction, while negative d_{trans} means the offset is to the left.

Summarising, as can be seen in Fig. 2, for two different positions, given the three values (s_{ID1} , d_{long1} , and d_{trans1}) or (s_{ID2} , d_{long2} , and d_{trans2}), the target position is completely defined.

The overall recursive procedure for tracking a target following a road segment is as follows: it receives the unbiased sensor measures (input measures), which are corrected with the estimated transversal offset obtained from the previous measures. These 2D offset-corrected measures would be over the road segment skeleton, if offset terms were perfectly estimated and there were no noise. As there is residual error in transversal offset estimator, and measures also suffer from noise, once the measure is offset-corrected, it gets projected (using a maximum likelihood approach) to the segment skeleton. This 1D projection is used as input measure of the IMM based longitudinal estimator, which tracks the traversed distance in this segment, and also its 1D velocity. In parallel, the lateral drift estimation is updated through a 1D Kalman

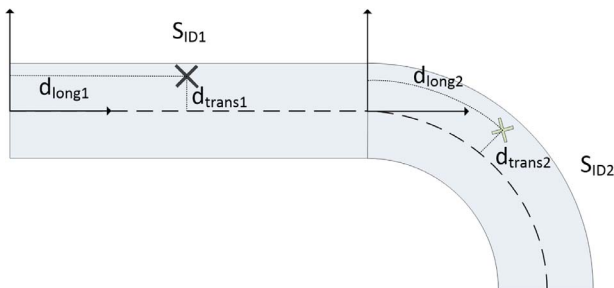


Fig. 2. Target position representation.

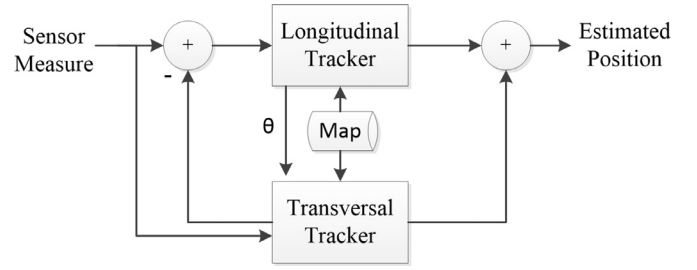


Fig. 3. Global filter structure.

filter whose input is the distance between the input measure and the road segment. Finally, both the updated lateral drift and the tracked longitudinal position are combined to obtain the final target kinematic state (position and velocity), taking into account the segment specification. This flow can be summarised in Fig. 3. Relating it with Fig. 2, the Longitudinal Tracking Filter tracks both d_{long} and its associated velocity, while the Transversal Filter tracks d_{trans} , using the target heading obtained by the longitudinal filter. Both filters are based on prediction models to be described in detail in Sections 5 and 7.

To complete the segment-adapted filtering description, a measurement error model must be established. The 2D measurement error is modelled as bias free, white, and following a bivariate normal distribution with potentially non constant covariance R_k (being k the number of measure). We also assume that the system has perfect measurement-to-track association.

The filter, as described up to now, would be able to track targets following an only road segment or an only road composed of several segments. To extend it to tracking on a road network we should not only know (d_{long} and d_{trans}), but also s_{ID} . The problem is near junctions, due to noisy measures and tracking, it is difficult quite often to make a decision on which is the value of s_{ID} . Our approach to do this is to maintain a list of the s_{ID} values in the network with an assessment of the associated probability of each s_{ID} . To keep the problem tractable, only the segments whose probability is above a certain threshold, P_{Hmin} , are kept in this collection. Those segments would be the ones in the vicinity of the last few measures, and compatible with preceding target movement. Therefore, using an Multiple Hypothesis Tracking (MHT) [27] inspired approach (the Map-Based Multiple Hypothesis Filtering or MBMHF approach introduced in Section 1), this list is updated along time with new segment hypotheses as the target arrives to junctions. Their relative probabilities are updated as new evidence is obtained based on the compatibility of measures with expected target movements, and finally the hypothesis list is pruned as the target leaves the junction and the segment followed by the target (and therefore its s_{ID}) becomes unambiguous. Then, a hypothesis ($H(j)$, with index j) is composed of four elements:

1. A segment identifier ($s_{ID}(j)$), specifying the segment ($S(j)$) where the target is assumed to be.
2. Another segment identifier ($s_{prevID}(j)$), specifying the segment ($S(j)$) where the target was before (the previous segment). The reason for including this second identifier has to do with dealing with ambiguities in the vicinity of segments junctions, as will be described in Section 6.
3. A 1D IMM based segment-adapted filter ($F(j)$), tracking d_{long} , its velocity and acceleration.
4. The probability of the hypothesis ($P(j)$).

After the definition of what we understand as a hypothesis, the general functional structure for the Longitudinal Tracking Filter will be clarified through the description of the items shown in the Fig. 4. The longitudinal filter makes use of a data base containing all the necessary information about the road segments (segment type and parameters),

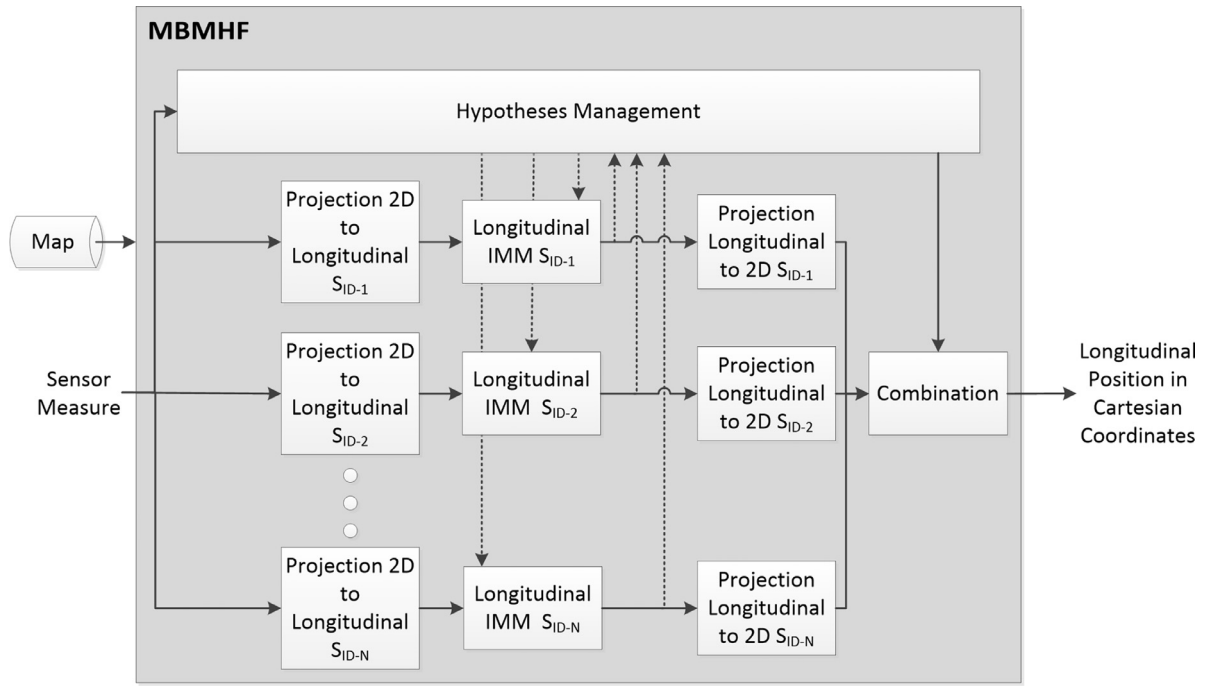


Fig. 4. Longitudinal estimator diagram.

used for locating and projecting the measures. The filter, following the MBMHF approach, keeps the list of hypotheses. After receiving a new measure, the filter projects it in the skeleton of all the segments in the vicinity of the segments included in collection of hypotheses. This projection is done in the functional blocks called Projection 2D to Longitudinal($s_{ID}(j)$). Then, the projected measure is processed by the segment-adapted filter ($F(j)$), called Longitudinal IMM($s_{ID}(j)$). The MBMHF has a key element, called in the figure Hypotheses Management, which manages all the hypotheses associated to the possible trajectories, creating new hypotheses as the target arrives to junctions, calculating the probability of each hypothesis, and pruning the collection of hypotheses (and the associated segment-adapted filters) as the probability of a given hypothesis becomes lower than P_{Hmin} . To calculate the probability of each hypothesis, it makes use of the kinematic state in all IMM filters in the hypotheses list ($\{H(j)\}$), the map and the measure, assessing its compatibility. IMM filters outputs are then projected back to 2D using segment geometric information (block Projection Longitudinal to 2D($s_{ID}(j)$) in the figure), and then they are combined using its probabilities. Section 6 describes this process in further detail.

Lateral drift tracking is much simpler. It should be noted this estimation is only performed when there is no ambiguity on the segment being followed ($\{H(1)\}$ is the only hypothesis in the list after pruning). As summarised in Fig. 5, it is decomposed in three consecutive processes. The first one is projecting the 2D measure in the transversal-to-segment direction, through the functional block Projection 2D to Transversal($s_{ID}(1)$), where 1 is the index of the only hypothesis. This projected measure of d_{trans} is used as the input for a Kalman filter in charge of estimating the piloting-induced drift. Then, the filtered d_{trans} estimate is projected back to 2D, using functional block Projection Transversal to 2D($s_{ID}(1)$). Additional details of the projections and the Kalman filter are given in Section 7.

Finally, note that after Longitudinal and Transversal trackers obtain their filtered estimates, both are transformed into 2D. Then, the output kinematic state can be obtained through the following additive process:

1. Position estimation is obtained through direct summation of 2D position estimates of both trackers.
2. Velocity estimation is directly Longitudinal tracker filtered estimate transformed to 2D, as there is no velocity component in the lateral tracker.
3. The output covariance (maybe needed for other outer algorithms, such as outlier detection or association) would be the addition of the longitudinal tracker covariance and the lateral tracker covariance (for 2D position). The lateral tracker does not add any covariance to velocity terms of the estimation.

4. Coordinate projection methods

In this section we are describing the different projection methods used in all parts of our filter. It should be emphasised not only position (and sometimes also velocity) should be projected, but also (measure or filter-calculated) covariances matrices should be projected, as our Bayesian filter approach needs them. It should be noted there are many different coordinate transformations, as there are three types of segments (straight lines, circular arcs and splines) and all the different types of projections needed: Projection from 2D to Longitudinal($s_{ID}(j)$) and Projection Longitudinal($s_{ID}(j)$) Kinetic State for longitudinal tracking; and Projection from 2D to Transversal($s_{ID}(j)$) and Projection Transversal to 2D($s_{ID}(j)$) for lateral drift tracking. In the next subsections we will cover all those coordinate transformations in a systematic way.

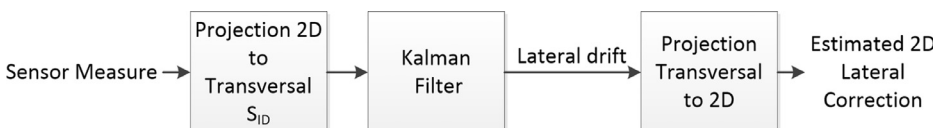


Fig. 5. Transversal estimator diagram.

4.1. Projection from 2D to longitudinal($s_{ID}(m)$)

In order to obtain the longitudinal projection from the 2D Cartesian Coordinates, we have implemented a maximum likelihood estimator (MLE). In our model, the measurement is modelled as:

$$\vec{z}_m = \vec{x}_r + \vec{\epsilon} \quad (2)$$

where $\vec{z}_m = [x_m \ y_m]^T$ is the 2D measure, $\vec{x}_r = [x \ y]^T$ is the 2D real position and $\vec{\epsilon} = [\epsilon_x \ \epsilon_y]^T$ is the 2D sensor measurement error, assumed to be zero-mean, Gaussian and time-uncorrelated. To obtain the MLE estimator, we constrain the possible longitudinal position of the target to the interest segment ($S(j)$, whose identifier is $s_{ID}(j)$), and therefore the 2D position results:

$$\hat{x}_{MLE} = \arg \max_{\vec{x}_r \in S(j)} p(\vec{z}_m | \vec{x}_r) = \max_{(x,y) \in S(m)} \frac{1}{2\pi |R_m|^{1/2}} \exp \left(-\frac{1}{2} \begin{bmatrix} x_m - x \\ y_m - y \end{bmatrix}^T \cdot R_m^{-1} \cdot \begin{bmatrix} x_m - x \\ y_m - y \end{bmatrix} \right) \quad (3)$$

where R_m is the sensor covariance, whose terms are:

$$R_m = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{pmatrix} \quad (4)$$

This problem is equivalent to minimise the Mahalanobis distance between the measure and the projection:

$$\hat{x}_{MLE} = \arg \min_{(x,y) \in S(m)} [x_m - x \ y_m - y] \cdot R_m^{-1} \cdot \begin{bmatrix} x_m - x \\ y_m - y \end{bmatrix} \quad (5)$$

Now, we have to particularise this expression to the three different segments: straight lines, circular arcs and splines. We do so by finding a suitable parametric definition of the segment and solving the aforementioned minimisation problem. Then, this projection has to be transformed into a distance to feed the filters, and also the measurement covariance need to be projected to obtain the distance measurement variance. In next subsections we will do this process for the different types of segments.

Finally, another interesting parameter, common to the three kind of segments, is the likelihood of the projection. It can be calculated as follows:

$$\lambda_{proj} = \frac{1}{2\pi |R_m|^{1/2}} \exp \left(-\frac{1}{2} [x_m - x_{proj} \ y_m - y_{proj}] \cdot R_m^{-1} \cdot \begin{bmatrix} x_m - x_{proj} \\ y_m - y_{proj} \end{bmatrix} \right) \quad (6)$$

where $(x_{proj} \ y_{proj})$ is the 2D result of the projection (in other words, the constrained MLE). This likelihood will be use in the calculation of the probability of each hypothesis in the MBMHF, as it is a probabilistic measure of the spatial compatibility between the 2D measurement and the associated segment.

4.1.1. Straight segments

In the case of the straight lines, the expression of the two coordinates of line can be written in term of an only parameter (x) as:

$$x = x; \quad y = m \cdot (x - x_0) + y_0 \quad (7)$$

where (x_0, y_0) is the initial point of the segment, and m is the slope. Substituting (7) in (5), we have the following equivalent 1D minimisation problem to find the X component of the MLE:

$$x_{proj} = \arg \min_x \begin{bmatrix} x - x_m \\ m \cdot (x - x_0) + y_0 - y_m \end{bmatrix}^T \cdot R_m^{-1} \cdot \begin{bmatrix} x - x_m \\ m \cdot (x - x_0) + y_0 - y_m \end{bmatrix} \quad (8)$$

Finding the minimum with respect to x , as the zero of the derivative of this function, it can be obtained:

$$x_{proj} = \frac{\sigma_y^2 x_m + \sigma_{xy} (y_0 - y_m - m(x_0 + x_m)) + \sigma_x^2 (m(y_m - y_0) + m^2 x_0)}{\sigma_y^2 - 2\sigma_{xy} m + \sigma_x^2 m^2} \quad (9)$$

Then, the 2D MLE results:

$$\hat{x}_{MLE} = \begin{bmatrix} x_{proj} \\ m(x_{proj} - x_0) + y_0 \end{bmatrix} \quad (10)$$

Finally, we have to transform the projected Cartesian coordinates into longitudinal ones. In order to do this, the distance from the beginning of the segment, which is the projected longitudinal distance, can be calculated with the following expression:

$$d_{long} = (x_{proj} - x_0) \sqrt{1 + m^2} \quad (11)$$

The previous equation can be rewritten applying (9) and collecting terms, as:

$$d_{long} = \left(\frac{(\sigma_y^2 - m\sigma_{xy})(x_m - x_0) + (m\sigma_x^2 - \sigma_{xy})(y_m - y_0)}{\sigma_y^2 - 2m\sigma_{xy} + m^2\sigma_x^2} \right) \sqrt{1 + m^2} \quad (12)$$

For the filter, it is also needed to calculate the longitudinal covariance of the projected longitudinal measure. As it is shown in (12) the projection from 2D measure to 1D projected longitudinal distance is a linear transformation, and therefore the exact longitudinal covariance can be obtained. After some algebra, it results:

$$\sigma_{long}^2 = \frac{(m^2 + 1)(\sigma_x^2 \sigma_y^2 - \sigma_{xy}^2)}{\sigma_y^2 - 2m\sigma_{xy} + m^2\sigma_x^2} \quad (13)$$

4.1.2. Circular arcs

Next the projection to the circular arc segment will be explained. Equivalently to the projection to a straight line, to project to a circular arc the Mahalanobis distance between the measure and the projection has to be minimised (according to (5)). The position in polar coordinates in a circular arc is given by:

$$x = x_c + R_c \cdot \cos(\varphi); \quad y = y_c + R_c \cdot \sin(\varphi) \quad (14)$$

where (x_c, y_c) is the centre of the curve, R_c is its radius and φ is the angle of the position with respect to X axis. Applying the previous equation in (5) we obtain the 1D minimisation problem:

$$\varphi_{proj} = \arg \min_{\varphi} \begin{bmatrix} x_c + R_c \cos \varphi - x_m \\ y_c + R_c \sin \varphi - y_m \end{bmatrix}^T R_m^{-1} \begin{bmatrix} x_c + R_c \cos \varphi - x_m \\ y_c + R_c \sin \varphi - y_m \end{bmatrix} \quad (15)$$

First, calculating the derivative of the function to be minimised with respect to φ , and equating to zero we obtain the following implicit definition of φ_{proj} :

$$A \cos \varphi_{proj} + B \sin \varphi_{proj} + C \cos \varphi_{proj} \sin \varphi_{proj} + D (\cos^2 \varphi_{proj} - \sin^2 \varphi_{proj}) = 0 \quad (16)$$

where:

$$\begin{aligned} A &= (y_c - y_m) P_{2,2} + (x_c - x_m) P_{1,2} \\ B &= -(x_c - x_m) P_{1,1} + (y_c - y_m) P_{1,2} \\ C &= R_c (P_{2,2} - P_{1,1}) \\ D &= R_c P_{1,2} \end{aligned} \quad (17)$$

with:

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} \\ P_{2,1} & P_{2,2} \end{pmatrix} = R_{med}^{-1} \quad (18)$$

Defining the sine in function of the cosine and substituting $\beta = \cos(\varphi_{proj})$ we obtain:

$$A\beta + B\sqrt{1 - \beta^2} + C\beta\sqrt{1 - \beta^2} + D(2\beta^2 - 1) = 0 \quad (19)$$

Regrouping:

$$A\beta + D(2\beta^2 - 1) = -\sqrt{1 - \beta^2}(B + C\beta) \quad (20)$$

Raising both sides of the equation to the square, and grouping terms we obtain a fourth degree polynomial:

$$a\beta^4 + b\beta^3 + c\beta^2 + d\beta + e = 0 \quad (21)$$

where

$$\begin{aligned} a &= C^2 + 4D^2 \\ b &= 4AD + 2BC \\ c &= A^2 - 4D^2 - C^2 + B^2 \\ d &= -2(AD + BC) \\ e &= D^2 - B^2 \end{aligned} \quad (22)$$

Resolving (21) and applying $\varphi_{proj} = \arccos(\beta)$, eight different possible solutions can be found. The solution is chosen by discarding the solutions not in the range of φ in the circular arc segment, and also by calculating the Mahalanobis distance for each solution (after converting the solution to Cartesian in 2D, calculating (x_{proj}, y_{proj}) from φ_{proj} using Eq. (14)), and selecting the minimum one. To translate the projected position angle into longitudinal, first the angle difference $\Delta\varphi$ with the start of the circular segment is calculated as:

$$\Delta\varphi = \varphi_{proj} - \varphi_0 \quad (23)$$

where φ_0 is the angle of the origin node of the curve:

$$\varphi_0 = \text{atan2}(y_0 - y_c, x_0 - x_c) \quad (24)$$

It should be noted that, depending on the rotation direction, the absolute value in (23) will keep the angle sign (counter-clockwise) or change it (clockwise). Of course, angle difference is protected against angle definitions discontinuities. Afterwards, this angular difference is translated into distance (with sign) as:

$$d_{long} = R_c * \Delta\varphi \quad (25)$$

In this case, the longitudinal covariance is obtained in the tangent of the projected position, in the same way that the previously explained for the straight line case, hence it is calculated with (13) and taking into account that the slope of the tangent is given by:

$$m = \frac{-1}{\tan(\varphi_{proj})} \quad (26)$$

4.1.3. Bézier curves

Finally, the projection to a Bézier curve will be explained. It should be remembered this kind of segments is defined as a collection of consecutive sections, each one defined using a parametric form described in (1). To obtain the projection into this kind of segment, projections on all sections are calculated, and then the one with minimum Mahalanobis distance is selected.

To calculate the projection on a given section (s_i), we use the parametric form of the curve according to (1) in (5), and therefore we obtain the minimisation problem, where the segment section subindex i has been discarded to reduce equations complexity.

$$\begin{aligned} \lambda_{proj} &= \arg \min_{\lambda} \left(\begin{matrix} a_{x1}\lambda^3 + a_{x2}\lambda^2 + a_{x3}\lambda + a_{x4} - x_m \\ a_{y1}\lambda^3 + a_{y2}\lambda^2 + a_{y3}\lambda + a_{y4} - y_m \end{matrix} \right)^T R_m^{-1} \\ &\quad \left(\begin{matrix} a_{x1}\lambda^3 + a_{x2}\lambda^2 + a_{x3}\lambda + a_{x4} - x_m \\ a_{y1}\lambda^3 + a_{y2}\lambda^2 + a_{y3}\lambda + a_{y4} - y_m \end{matrix} \right) \end{aligned} \quad (27)$$

Calculating the derivative and equating to zero the function to be minimized, a quintic function in λ , is obtained.

$$A\lambda_{proj}^5 + B\lambda_{proj}^4 + C\lambda_{proj}^3 + D\lambda_{proj}^2 + E\lambda_{proj} + F = 0 \quad (28)$$

with:

$$\begin{aligned} A &= 6(a_{x1}^2\sigma_x^2 + a_{y1}^2\sigma_y^2 + 2a_{x1}a_{y1}\sigma_{xy}) \\ B &= 10(a_{x1}a_{x2}\sigma_x^2 + (a_{x1}a_{y1} + a_{x2}a_{y1})\sigma_{xy} + a_{y1}^2\sigma_y^2) \\ C &= 4(2a_{x1}a_{x3} + a_{x2}^2)\sigma_x^2 + 8(a_{x1}a_{y1} + a_{x2}a_{y1} + a_{x3}a_{y1})\sigma_{xy} + 12a_{y1}^2\sigma_y^2 \\ D &= 6(a_{x1}a_{x4} - a_{x1}x_m + a_{x2}a_{x3})\sigma_x^2 + 6(2a_{y1}^2 - a_{y1}y_m)\sigma_y^2 \\ &\quad + 6(a_{x1}a_{y1} - a_{x1}y_m + a_{x2}a_{y1} + a_{x3}a_{y1} + a_{x4}a_{y1} - a_{y1}x_m)\sigma_{xy} \\ E &= 2(2a_{x2}a_{x4} - 2a_{x2}x_m + a_{x3}^2)\sigma_x^2 + 2(3a_{y1}^2 - 2a_{y1}y_m)\sigma_y^2 \\ &\quad + 4(a_{x2}a_{y1} - a_{x2}y_m + a_{x3}a_{y1} + a_{x4}a_{y1} - a_{y1}x_m)\sigma_{xy} \\ F &= 2(a_{x3}a_{x4} - a_{x3}x_m)\sigma_x^2 + 2(a_{x3}a_{y1} - a_{x3}y_m + a_{x4}a_{y1} - a_{y1}x_m)\sigma_{xy} \\ &\quad + 2(a_{y1}^2 - a_{y1}y_m)\sigma_y^2 \end{aligned} \quad (29)$$

It can be solved using numerical methods, such as the Newton's method. As in each section λ_{proj} must be between 0 and 1, the projection is passed through a non linearity to ensure the projection is compatible with segment definition:

$$\lambda_{proj} = \begin{cases} 0 & \lambda_{proj} < 0 \\ 1 & \lambda_{proj} > 1 \\ \lambda_{proj} & \text{otherwise} \end{cases} \quad (30)$$

Afterwards, the λ_{proj} parameter is introduced in Eq. (1) for each section of the Bézier curve, to get the 2D projection in each of the sections defining the segment $(x_{proj,i}, y_{proj,i})$. In order to identify the projection in the whole segment, the projection $(x_{proj,i}, y_{proj,i})$ with lowest Mahalanobis distance to the measure (x_m, y_m) is selected. This allows to identify both the section s_i and the λ_{proj} univocally defining the 2D position of the projection in the whole Bézier curve segment (x_{proj}, y_{proj}) .

The next step is to transform the projected λ_{proj} into longitudinal coordinates (along segment distances). Using Composite Simpson's rule, the distance in the section s_i would result:

$$d_{long,i}(\lambda_{proj}) = \int_0^{\lambda_{proj}} \sqrt{\left(\frac{\partial x}{\partial \lambda}\right)^2 + \left(\frac{\partial y}{\partial \lambda}\right)^2} d\lambda \quad (31)$$

This approach demands resolving a numerical integration for each measure projection, due to the lack of a simple expression that relates distance with λ , resulting in high computational load. To reduce this load our proposal is sampling Eq. (31) offline, and approximating it with a polynomial. We use a polynomial curve fitting of degree 4, which our tests have shown be capable of providing a good enough fit of the aforementioned equation. So we have:

$$d_{long,i}(\lambda_{proj}) \approx p_{1,i}\lambda_{proj}^4 + p_{2,i}\lambda_{proj}^3 + p_{3,i}\lambda_{proj}^2 + p_{4,i}\lambda_{proj} + p_{5,i} \quad (32)$$

As the longitudinal distance is zero when λ is zero, the coefficient $p_{5,i}$ has to be zero. The full distance from the beginning of the Bézier curve can be obtained adding the length of the previous sections (up to a total of $i - 1$). As these sections whole length can be calculated from their respective distance interpolation function (Eq. (32)), for $\lambda = 1$, the complete distance can be calculated as:

$$d_{long} = \sum_{k=1}^{i-1} d_{long,k}(1) + d_{long,i}(\lambda_{proj}) \quad (33)$$

The longitudinal covariance of the projection is obtained in the tangent of the projected position, in a similar way that the previously explained for the straight line case, hence it is calculated again using (13), knowing that the slope of the point is given by:

$$m = \frac{\partial y}{\partial x} = \frac{\partial y}{\partial \lambda} \cdot \frac{\partial \lambda}{\partial x}^{-1} \quad (34)$$

From (1), taking derivatives, and using the parameters of the section where the projection lies, we have:

$$\begin{aligned}\frac{\partial x}{\partial \lambda} &= 3a_{x1,i}\lambda^2 + 2a_{x2,i}\lambda + a_{x3,i} \\ \frac{\partial y}{\partial \lambda} &= 3a_{y1,i}\lambda^2 + 2a_{y2,i}\lambda + a_{y3,i}\end{aligned}\quad (35)$$

To calculate the slope in the projection position we substitute $\lambda = \lambda_{proj}$ in previous equations, resulting:

$$m = \frac{3a_{y1,i}\lambda_{proj}^2 + 2a_{y2,i}\lambda_{proj} + a_{y3,i}}{3a_{x1,i}\lambda_{proj}^2 + 2a_{x2,i}\lambda_{proj} + a_{x3,i}} \quad (36)$$

4.2. Projection from longitudinal($s_{ID}(m)$) to kinetic state

The different along segment tracking filters estimate target position and velocity in terms of the along-segment tracking estimation vector. This 2D vector contains the distance along the segment (d_{long}), and its associated time derivative; the longitudinal velocity ($v_d = \frac{dd_{long}}{dt}$). The along-segment tracking estimation vector (\vec{d}) then results:

$$\vec{d} = [d_{long} \quad v_d]^T \quad (37)$$

From this data, the process to obtain the full kinetic state ($\vec{\Theta}$) of a target (i.e. predicted or filtered estimates) will be described next. This process depends on the type of segment, although the output kinematic state of the process always contains the same components: X and Y Cartesian position, groundspeed v_{gs} and target heading (θ , defined as angle with Y axis, in clockwise direction):

$$\vec{\Theta} = [x \quad y \quad v_{gs} \quad \theta]^T \quad (38)$$

X and Y Cartesian coordinates are calculated from d_{long} and from the parametric definition of the segments, and therefore it is dependent on the type of segment. Meanwhile, it is clear v_d modulus is directly v_{gs} :

$$v_{gs} = |v_d| \quad (39)$$

Regarding target heading θ , the method to actually derive it is slightly different for each kind of segment, and will be described in next subsections.

To assess filter output variance it is also important to calculate $\vec{\Theta}$ estimation covariance for each along-segment tracking filter. This covariance can be related to the 2x2 covariance matrix of the error of the along segment tracking estimation vector \vec{d} , which has the following form:

$$P_{long} = \begin{pmatrix} \sigma_{long}^2 & \sigma_{long-vel} \\ \sigma_{long-vel} & \sigma_{vel}^2 \end{pmatrix} \quad (40)$$

The projection from longitudinal to kinematic state can be approximated by a linear transformation, and therefore the approximate covariance of $\vec{\Theta}$ (P_S) can be obtained. After some algebra, it results:

$$P_S = \begin{pmatrix} \frac{\sigma_{long}^2}{m_{local}^2 + 1} & \frac{m_{local}\sigma_{long}^2}{m_{local}^2 + 1} & \frac{\sigma_{long-vel}}{\sqrt{m_{local}^2 + 1}} & 0 \\ \frac{m_{local}\sigma_{long}^2}{m_{local}^2 + 1} & \frac{m_{local}^2\sigma_{long}^2}{m_{local}^2 + 1} & \frac{m_{local}\sigma_{long-vel}}{\sqrt{m_{local}^2 + 1}} & 0 \\ \frac{\sigma_{long-vel}}{\sqrt{m_{local}^2 + 1}} & \frac{m_{local}\sigma_{long-vel}}{\sqrt{m_{local}^2 + 1}} & \sigma_{vel}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\theta}^2 \end{pmatrix} \quad (41)$$

where m_{local} is the slope of the curve and σ_{θ} is a default value for the track angle covariance. It should be noted m_{local} is related with target heading θ as follows:

$$m_{local} = \frac{1}{\tan \theta} \quad (42)$$

Again, to fully specify the transformation of both states and covariances, the following procedures need to be defined:

1. How to obtain XY coordinates from d_{long}
2. How to obtain the heading from d_{long} , v_d and curve parameters.

This will be detailed in next subsections for each type of segment.

4.2.1. Straight segments

In the case of the straight lines the calculation of X and Y coordinates can be directly made by inverting (11), resulting in the following expressions:

$$x = x_0 + \frac{d_{long}}{\sqrt{1 + m^2}} \quad (43)$$

$$y = y_0 + \frac{m \cdot d_{long}}{\sqrt{1 + m^2}} \quad (44)$$

The heading can be obtained as follows:

$$\theta = \arctan(m^{-1}) + \pi \cdot u(-v_d \cdot m) \quad (45)$$

where $u(x)$ is the Heaviside step function. The reason v_d is negative means the movement goes in descending X direction, and therefore a π constant need to be added to the arctangent. It is clear that, in this case, combining Eqs. (45) and (42) the slope m_{local} remains constant and equal to m .

4.2.2. Circular arc

In the case of the circular arc, the measure transformation from longitudinal to Cartesian coordinates is made with the following procedure:

$$\Delta\varphi_{abs} = \frac{d_{long}}{R_c} \quad (46)$$

where $\Delta\varphi_{abs}$ is the signed value of the rotation angle difference on the curve. To calculate x and y we have:

$$x = x_c + R_c \cdot \cos(\varphi_0 + \Delta\varphi_{abs}) \quad (47)$$

$$y = y_c + R_c \cdot \sin(\varphi_0 + \Delta\varphi_{abs}) \quad (48)$$

The heading can be calculated with the following expression:

$$\theta = -(\varphi_0 + \Delta\varphi_{abs}) + \pi \cdot u(-v_d) \quad (49)$$

where $u(x)$ is again the Heaviside step function and the term v_d is used to check if the rotation is counter-clockwise rotation or clockwise and changes the heading accordingly.

4.2.3. Bézier curve

In the Bézier curve case, the measure transformation from d_{long} to Cartesian Coordinates is made obtaining λ from d_{long} and then substituting the resulting λ in (1). The first process has to be done in several steps:

1. Identify the relevant Bézier curve section s_i . This is done by successively comparing d_{long} with the accumulated length of curve sections, and finding the first section whose accumulated length is larger than d_{long} . From the description of (33), and remembering the definition of d_{long} , $k(\lambda)$ in (32), the accumulated length of i th curve section (s_i) is:

$$d_{acum}(i) = \sum_{k=1}^i d_{long,k}(1) \quad (50)$$

2. Obtain the remaining distance $d_{long,i}(\lambda)$ within this section from d_{long} , subtracting the accumulated length of the previous curve section (s_i):

$$d_{long,i}(\lambda) = d_{long} - d_{acum}(i-1) \quad (51)$$

3. Solve the implicit 4th degree equation relating the previously calculated $d_{long,i}(\lambda)$ and the needed λ :

$$p_{1,i}\lambda^4 + p_{2,i}\lambda^3 + p_{3,i}\lambda^2 + p_{4,i}\lambda + p_{5,i} - d_{long,i}(\lambda) = 0 \quad (52)$$

Finally, the resulting λ is used in Eq. (1) to obtain X and Y coordinates.

The heading is obtained using the local slope of the point (see Eq. (35)), resulting:

$$\theta = \text{atan2}(3a_{x1,i}\lambda^2 + 2a_{x2,i}\lambda + a_{x3,i}, 3a_{y1,i}\lambda^2 + 2a_{y2,i}\lambda + a_{y3,i}) + \pi \cdot u(-v_d) \quad (53)$$

where $u(x)$ is again the Heaviside step function, used to change heading for movements with decreasing distance (negative v_d).

4.3. Projection from 2D to transversal($s_{ID}(m)$)

In order to estimate the lateral drift, the sensor measure has to be transformed also to a measure of the distance between the segment s_{ID} and the target (d_{trans}). Again, this calculation is different for different types of segments. Also, for the later filter processing it is important to calculate the associated error covariance (to be called σ_{trans}).

To calculate the distance to the segment we approximate it by the distance from the point to the line tangent to the projection calculated in Section 4.1. The distance sign is positive to the right of target movement direction and is negative to the left. Using the equation of distance between point and segment, and the target heading described in Section 4.2, it results:

$$d_{trans}(m) = \cos \theta \cdot (x_m - x_{proj}) - \sin \theta \cdot (y_m - y_{proj}) \quad (54)$$

It should be noted that this projection uses the target heading θ , and therefore it is made after longitudinal filtering. The index (m) in $d_{trans}(m)$ is used to refer to the measured lateral drift.

With the Eq. (42) that relates θ with m_{local} and assuming the impact of the error in the calculation of m_{local} , x_{proj} and y_{proj} from x_m and y_m in the error of $d_{trans}(m)$ is negligible (which is true for straight segments, and also for curves with not too small local radius), we can derive the transversal covariance as:

$$\sin \theta = \frac{1}{\sqrt{1+m_{local}^2}} \quad \cos \theta = \frac{m_{local}}{\sqrt{1+m_{local}^2}} \quad (55)$$

$$\sigma_{trans}^2 = \frac{\sigma_x^2 m_{local}^2 + \sigma_y^2 - 2\sigma_{xy} m_{local}}{m_{local}^2 + 1} \quad (56)$$

4.4. Projection from transversal($s_{ID}(m)$) to 2D offset

To convert lateral drift tracking results into 2D, in order to add the results with those of longitudinal tracking, as depicted in Fig. 3, the lateral drift d_{trans} must be projected to Cartesian coordinates, using the same sign convention in Section 4.3. The calculation is based on calculating the vector in the direction orthogonal to the heading. This heading is calculated following the methods described in Section 4.2, calculated differently for each type of segment. From this result, the Cartesian 2D offset vector, defined as $[\Delta x, \Delta y]^T$, can be calculated as:

$$\Delta x = d_{trans}(f) \cos \theta \quad (57)$$

$$\Delta y = -d_{trans}(f) \sin \theta \quad (58)$$

Here, The index (f) in $d_{trans}(f)$ is used to refer to the filtered lateral drift. Again, in order to estimate the overall filter output covariance it is necessary to define a covariance term (R_{drift}) associated to the Cartesian 2D offset vector. From Eqs. (55), (57) and (58), and assuming lateral drift estimator has an standard deviation equal to σ_{trans} , we get:

$$R_{drift} = \frac{1}{m_{local}^2 + 1} \begin{pmatrix} m_{local}^2 \sigma_{trans}^2 & -m_{local} \sigma_{trans}^2 \\ -m_{local} \sigma_{trans}^2 & \sigma_{trans}^2 \end{pmatrix} \quad (59)$$

5. Along-segment IMM filter description

After describing the coordinate projections, we will focus in the longitudinal (along-segment) filter design. The longitudinal movement along a given segment is modelled using the following simplified state vector:

$$x_{long} = (d_{long}, v_{long}, a_{long})^T \quad (60)$$

where d_{long} is the signed distance measured from the segment origin node (calculated as described in previous section), v_{long} is the signed longitudinal velocity at which the target moves away from the origin node, and a_{long} is the signed acceleration of the target in this direction. To track the movement along each segment (associated to an hypothesis within the MBMHF) is estimated by a longitudinal IMM filter [7]. The designed IMM filter uses two Kalman filters defined by two dynamic motion models. The first one is in charge of segments without longitudinal accelerations, so it models a constant velocity movement with the following state variables:

$$x_1 = (d_{long}, v_{long})^T \quad (61)$$

The second mode assumes constant longitudinal acceleration, and therefore has the following state variables:

$$x_2 = (d_{long}, v_{long}, a_{long})^T \quad (62)$$

This second filter also has higher process noise in order to be able to follow not fully modelled manoeuvres, using plant noise model as the indicated by the following equation:

$$Q_{KA} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_{KA}^2 T^2 \end{pmatrix} \quad (63)$$

where σ_{KA} is the comfort standard deviation. This plant noise is derived from Singer model assuming that the target suffer small random perturbations [28]. The parameters used to model the transition probabilities are those proposed in for a two mode structure [26].

The measurements feeding the IMM filter are those obtained, after lateral-drift removal, using the projection processes described in Section 4.1. In that section also the measurement covariances derivation was described.

The filter tracks three states, but a 2D vector only containing distance and velocity is provided as output and finally transformed to kinetic state coordinates according to the procedures in Section 4.2. Also, the filter covariance is of size 3x3, but only the 2x2 submatrix related to distance and velocity estimates is provided as output and further processed.

An important parameter to be estimated for the MBMHF is the IMM filter likelihood. This likelihood is a measure of the compatibility of longitudinal projected measures with the prior IMM state. In this paper is proposed to use the likelihood of the constant speed filter as the IMM likelihood. It can be calculated as follows:

$$\lambda_{IMM} = \frac{1}{2\pi|\hat{S}|^{1/2}} \exp\left(-\frac{1}{2} \cdot res^T \cdot \hat{S}^{-1} \cdot res\right) \quad (64)$$

where res is the measurement residual and \hat{S} is its covariance.

6. Map-Based multiple hypothesis filtering

Once the road map representation, the measure projection process and the along-segment tracker have been described, the longitudinal tracking (the MBMHF) is going to be defined. In a general way the tracking system starts associating the sensor measure with the close

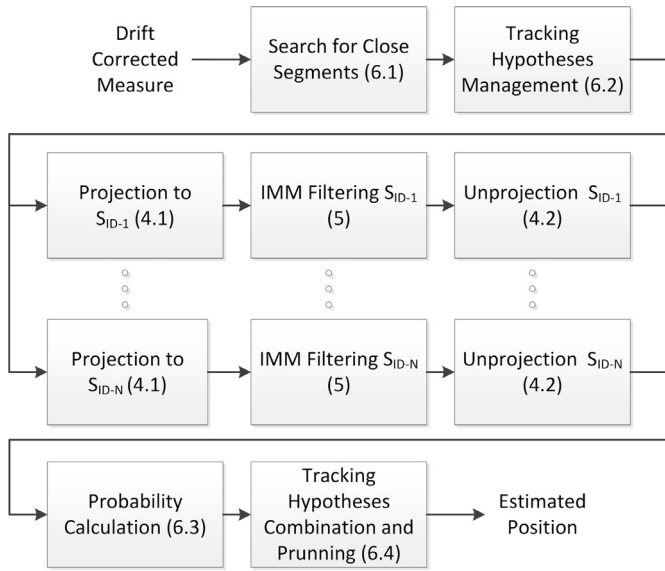


Fig. 6. Workflow of the MBMHF.

segments (process to be outlined in Section 6.1), and managing the associated tracking hypothesis (as will be described in Section 6.2). This is followed by projecting the (lateral drift-corrected) measure to each of these segments using the methods described in Section 4.1. Later, a longitudinal IMM estimator (described in Section 5) is used to smooth the longitudinal distance measurement. Afterwards, each of the potential hypothesis probabilities are calculated (following the method to be described in Section 6.3). Finally, each hypothesis is unprojected into Cartesian coordinates (as outlined in Section 4.2) and they are combined using the hypotheses probabilities, as will be described in Section 6.4. Also, very low probability hypotheses get pruned. The whole process is summarised in Fig. 6.

6.1. Search for close segments

As previously explained, the measures need to be projected to the nearby segments. When the first measure is received the distance to all the segments of the airport is calculated and only if the distance is below a certain threshold the segment is included in the list of potential target localizations. This procedure can be alleviated using a spatial cell ordering of the segments. For later measures, the close segments list is obtained from the close-segments list of the previous measure associated to this track. They will either be the same segments or segments adjacent to the ones in the previous localization process. Again, all segments in the list of close segments need to be at a distance from the measure lower than a predefined threshold (th_{loc}), higher than the expected measurement error but not too high to keep computational load low.

6.2. Tracking hypotheses management

The multiple hypotheses management process within the MBMHF is in charge of associating close segments list with previous hypotheses. New hypotheses will be created in ambiguity situations as for example intersections. Each hypothesis (with index j) assumes the target moves in a specific segment so it is formed, as previously described, by: 1) the current segment identifier ($s_{ID}(j)$); 2) the previous segment identifier ($s_{prevID}(j)$); 3) the 1D IMM based segment-adapted filter ($F(j)$); and 4) the probability of the hypothesis ($P(j)$).

The MBMHF is initialised at a moment where the localization is unambiguous (close-segment list has only one element), so it is assumed that the target is known to be in an specific segment. This way, it is created the hypothesis associated to that location, with both $s_{ID}(1)$ and

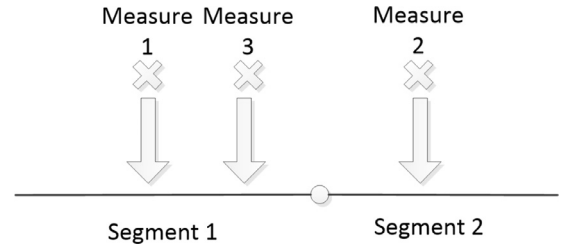


Fig. 7. Effect of the sensor noise.

$s_{prevID}(1)$ equal to that segment, the $F(1)$ filter initialised with an only measure (assuming velocity and acceleration to be zero, but with extremely high associated variances), and hypothesis probability $P(1)$ equals to 1. This simplified initialisation is enough for many applications where targets enter a road network area by a set of well-separated defined roads. Of course, alternative initialisations could be easily implemented.

With the following projected measures, there are different options:

1. A localization (close-segment derived in 6.1, and included in the close-segments list) matches with an existing hypothesis $s_{ID}(j)$ parameter. In this case, the localization is simply associated with the hypothesis, which will be updated with this measurement. So in this case, $s_{ID}(j)$, $s_{prevID}(j)$, $F(j)$, and $P(j)$ remain unmodified.
2. The localization matches with the previous segment of an existing hypothesis. It is important to take into account this case in order to prevent situations as the one shown in Fig. 7. In that example, measures 1, 2 and 3 are time ordered, and the target moves quite slowly through segment 1. Because of the sensor noise measure 2 is associated with segment 2 when the target is still moving in the previous segment. Measure 3 projections returns to segment 1 but the hypothesis associated to the segment 1 has already evolved to segment 2. In this case the localization is associated with the hypothesis, but after the projection to $s_{prevID}(j)$ the resulting d_{long} need to be modified adding/subtracting the length of the implicated segments, making its distance measured from the current hypothesis segment $s_{ID}(j)$ (segment 2 in the example). In this case, $s_{ID}(j)$, $s_{prevID}(j)$, $F(j)$, and $P(j)$ also remain unmodified.
3. The localization does not match with any created hypothesis $s_{ID}(j)$ parameter but it is a valid transition from one of them (there is a junction between a previous hypothesis $s_{ID}(j)$ segment and the localization one). In this case a new hypothesis is created with the following parameters: $s_{ID}(j)$ will be the localization close-segment; $s_{prevID}(j)$ will be the previous hypothesis $s_{ID}(j)$ value; $F(j)$ state values are modified to the new segment (d_{long} gets added/subtracted the length of the implicated segments, and the velocity and accelerations may change the sign depending on the orientations of both segments); and $P(j)$ is maintained.
4. Finally, in case that the localization does not match with any of the previously explained cases, the measure is discarded assuming that it is a sensor outlier.

The previous logic is summarised in Fig. 8.

6.3. Tracking hypotheses probability calculation

Once the phase of associate locations with hypotheses finishes, the system makes projections to all hypothesis segments, the IMM filters are updated with the new projected measures, and the longitudinal state gets unprojected into kinetic state vector coordinates as described in 4.2. After this, the probability of each hypothesis is calculated. This probability is obtained taking into account the following parameters: the hypothesis probability according to prior information ($P(j)$) value, the projection likelihood (as described in Section 4.1) and the IMM

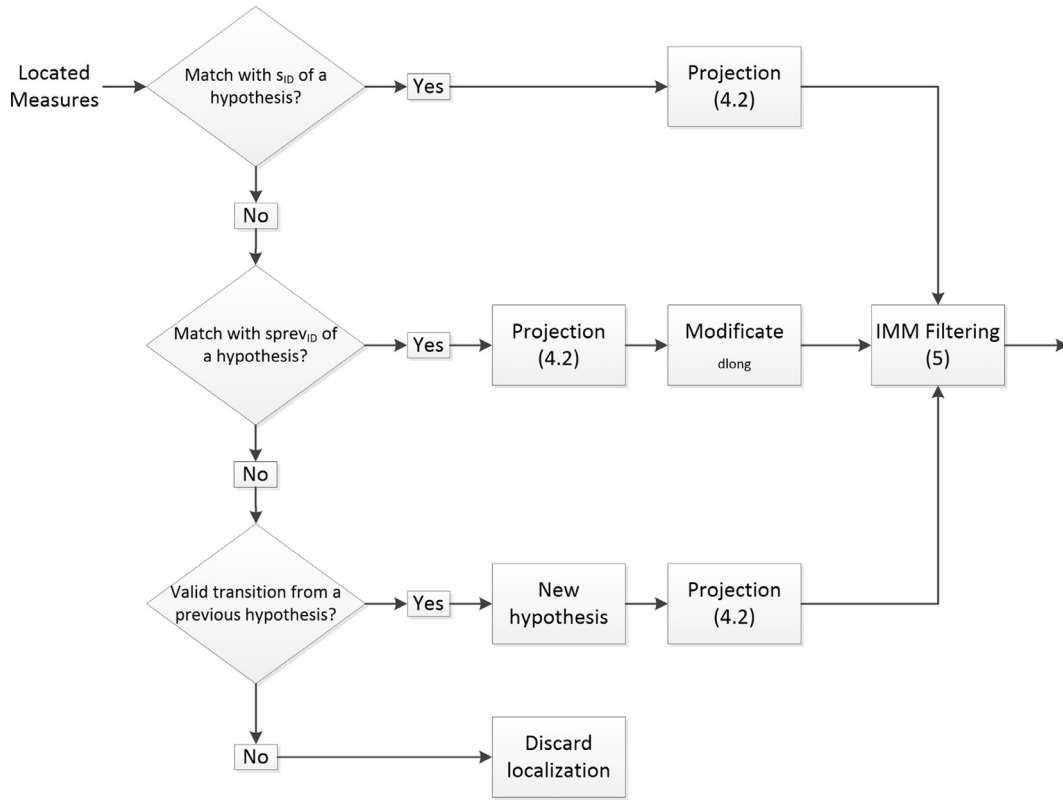


Fig. 8. Flow diagram of the hypotheses update.

likelihood (described in Section 5). The higher the projection likelihood the most compatible the 2D measure is with the segment, and the higher the IMM likelihood the most consistent the segment filter is with the projected longitudinal measures. Hence, neglecting dependency between those to probabilistic measure, the following expression (obtained from the Bayes' theorem assuming likelihood and prior probabilities independence) is used to update the probabilities:

$$P(j) = \frac{P(j) \cdot \lambda_{proj}(j) \cdot \lambda_{IMM}(j)}{\sum_{s=1}^M P(s) \cdot \lambda_{proj}(s) \cdot \lambda_{IMM}(s)} \quad (65)$$

where $P(j)$ is the probability (posterior at the left of the equation, prior at the right), $\lambda_{proj}(j)$ is the projection likelihood (see Eq. (6)), $\lambda_{IMM}(j)$ is the filter likelihood (see Eq. (64)). All these terms are calculated for the same hypothesis with index j the hypothesis index. Additionally, M is the number of updated hypotheses. Obviously, the calculation includes a probability normalisation, so that all hypothesis probabilities sum one.

Before calculating the probabilities according to (65), very unlikely projections are discarded using the following procedure: the calculated likelihood of each projection ($\lambda_{proj}(j)$) is compared to a threshold (calculated using chi-square tables for two degrees of freedom, $\lambda_{proj_{max}}$) in order to discard the not likely ones. If this is the case, the hypothesis would be in the same case as if it had no close-segment associated for the current measurement. Additionally, the associated IMM filter ($F(j)$) would need to go back to the estimates available before the current measure was processed.

If any hypothesis has not been updated with the last measure but it has not been pruned, the projection likelihood is the same as the projection threshold and the filter likelihood is the minimum of the updated hypothesis divided by a constant ($N_{like_{miss}}$).

6.4. Tracking hypotheses combination and pruning

After each hypothesis has its probability calculated and the

coordinates transformed, the MBMHF combines the unprojected kinetic state vectors for each hypothesis to obtain the final output with a probability-weighted average. If any hypothesis has not been updated with the last measure, its estimation is predicted to the current time assuming constant velocity.

Before this combination, an optional heading protection can be applied for low velocities. As the hypothesis are formed by the current segment location ($s_{ID}(j)$) and the previous one ($s_{prevID}(j)$), the heading can be obtained by the transition sense between them. This protection allows to estimate correctly the heading avoiding jumps of π radians when the target is stopped. It is important to remark that this should be only used in scenarios where the targets are not expected to perform a sharp change of direction without the use of multiple segments, e.g. airports.

Finally, the output kinetic state estimate of the MBMHF would result:

$$\vec{\Theta}_{MBMHF} = \sum_{i=1}^M P(i) \cdot \vec{\Theta}(i) \quad (66)$$

where $\vec{\Theta}(i)$ is the estimated kinetic state of the hypothesis with index i , calculated as described in Section 4.2. It should be noted the fourth component of the state is an angle in radians, and therefore this average must be protected against angle discontinuities.

The associated output covariance (P_{MBMHF}) is obtained with the following expression of the covariance of the probabilistic mix of random vectors:

$$P_{MBMHF} = \sum_{i=1}^M P(i) \cdot \left[P_{\Theta}(i) + (\vec{\Theta}(i) - \vec{\Theta}_{MBMHF})(\vec{\Theta}(i) - \vec{\Theta}_{MBMHF})^T \right] \quad (67)$$

where $P_{\Theta}(i)$ is the estimated covariance of the hypothesis i , calculated according to the procedures described in 4.2.

Finally, those hypotheses that have not been updated for the last few measures get removed from the hypothesis list (T_{max}).

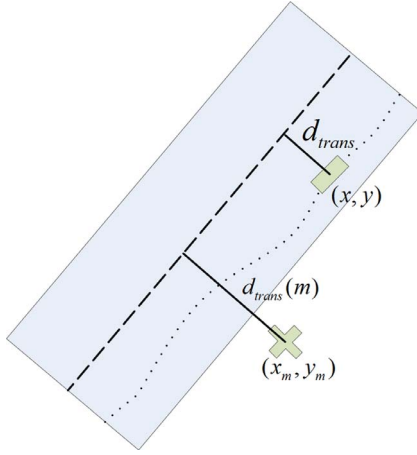


Fig. 9. Lateral drift estimation.

7. Lateral drift tracker

As described previously, the system also tracks the transversal movement (lateral drift) following the distance between the target and the centre of the road: as can be seen in Fig. 9, the humane-piloted (or automatic-piloted) target will, in general, not perfectly follow the road axis.

In this figure the 2D position of the target (x, y) and the 2D measurement vector (x_m, y_m) are depicted. Also, the actual target lateral drift (d_{trans}) is depicted, and also the drift measurement ($d_{trans}(m)$). To obtain the filtered drift estimation ($d_{trans}(f)$) it is assumed drift dynamic behaviour can be modelled as a slowly changing signal. In our implementation this is done through a one dimensional vector state Kalman filter, with an exponential time-correlation, although higher order movement models could be implemented with almost no changes in the rest of the complete filtering architecture. So, this filter state vector is:

$$x_{trans} = (d_{trans}) \quad (68)$$

This model can be expressed as follows, where $[k]$ is the time index.

$$d_{trans}[k] = \rho \cdot d_{trans}[k-1] + w[k] \quad (69)$$

In this model, ρ is a constant related to the decorrelation time of the process and measurement time and $w[k]$ is a stochastic discrete white process with normal distribution of zero mean and variance equals to $\sigma_{drift}^2(1 - \rho^2)$. This model assumes that in stationary state d_{trans} has a normal distribution of zero mean and variance equals to σ_{drift}^2 . This estimation is obtained with a Kalman Filter with only one state, following this prediction model and assuming measurement error ($v[k]$) have zero mean and variance equal to σ_{trans}^2 , as calculated according to Eq. (56). The measurement model then results:

$$d_{trans}(m)[k] = d_{trans}[k] + v[k] \quad (70)$$

The lateral drift is only calculated in time intervals where the segment being followed by the target is not ambiguous (MBMHF has only one hypothesis), although it is corrected at all times but at MBMHF input (to reduce the impact of lateral drift on hypothesis formation and management), and after MHMBF, as the drift is part of the actual state of the target. So, once the offset is calculated, the sensor measures are corrected with it ($d_{trans}(f)$ is translated to $(\Delta x, \Delta y)$ and this vector is subtracted from (x_m, y_m) before localization and projection) so the

Table 1
Sensor characteristics.

	SMR	Description
σ_p	5 m	Range standard deviation
σ_θ	0.2°	Azimuth standard deviation
T_s	1 s	Update rate
Position	[-100,100] m	Cartesian coordinates

MBMHF localization performance is not deteriorated. Once the longitudinal map-constrained position of target has been estimated by the MBMHF, the 3D drift $(\Delta x, \Delta y)$ is added again so the position estimation (first two elements of the MBMHF state vector \vec{S}_{MBMHF} , calculated as described in Eq. (66)) does not get biased in relation to the real target position. Fig. 10 outlines the described process.

8. Simulation and experimental results

In order to show the performance of our filter and compare it with competing systems in the literature we will analyse two scenarios of an aircraft moving along the airport surface, one with real data and other with simulated data. It is important to remind that our filter assumes that the measures from different targets are already associated and bias free.

Also, to complete the results section, we have analysed the performance of our filter and that of competing systems in terms of computational load.

8.1. Simulation results

The simulated data is obtained emulating a Surface Movement Radar SMR whose characteristic are shown in the Table 1.

The trajectory represents a target moving by the airport surface, using a flat earth model and 2D simulation, enough for our analysis. At the beginning, the target is stopped at [100,300] and it starts to accelerate to a speed of 7.5 m/s. Then, it moves at constant velocity until it decelerates and keep stopped during 25 s. After that, the target performs another acceleration to 15 m/s. At the end, the target stops again. Table 2 and Fig. 11 summarise this trajectory as a set of longitudinal acceleration segments and through the ideal path on the road segments.

Table 3 shows the constants used in our system. Each constant has the corresponding section in which it has been explained.

Fig. 12 shows some results regarding to the localization and projection of a set of measurements of a target following part of the described trajectory. In this case, only the longitudinal projection is depicted, the target has no lateral drift, and the lateral drift tracker is disabled ($d_{trans}(f)$ is set to zero). It should be noted the along-segment projection is not normal, it is affected by the measurement covariance matrix, and therefore is different for each segment in the map and position.

Next, Fig. 13 shows a projection to multiple segments. The target is moving and it arrives to an intersection with three options. The system projects the measures to the three roads until two of the segments are too far from the measures. After a little time, the associated hypotheses would be pruned. Again, in this case, the target has no lateral drift, and the lateral drift tracker is disabled.

Next some one-run results concerning the longitudinal tracking are presented. Fig. 14 shows the result of the complete longitudinal tracking system in an intersection point. The target, coming from the right of the figure, is turning to its left and with the first measures the



Fig. 10. Lateral drift correction.

Table 2
Trajectory description.

Time [s]	Speed [m/s]	Longitudinal Acceleration [m/s^2]
0–15	-	0.5
15–75	7.5	0
75–90	-	-0.5
90–115	0	0
115–145	-	0.5
145–180	15	0
180–210	-	-0.5
210–240	0	0

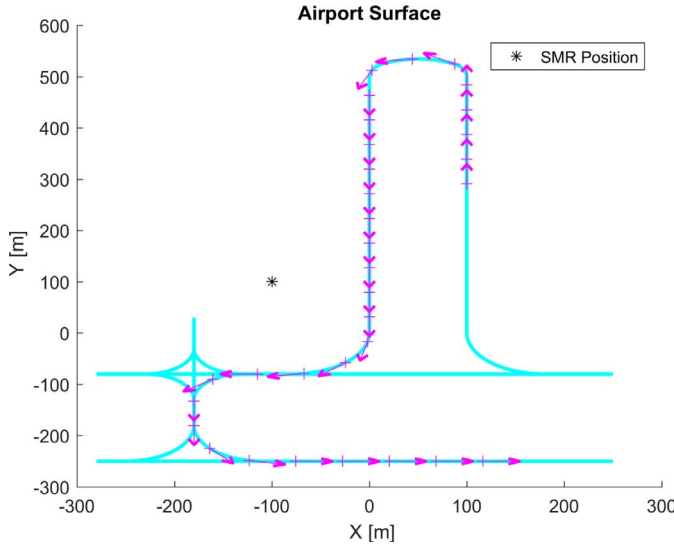


Fig. 11. Ideal trajectory.

Table 3
MBMHF Constants values.

Name	Symbol	Value	Section
Kalman Comfort Deviation	σ_{KA}	0.01 m/s^3	5
Segment Search Threshold	\bar{t}_{loc}	$5 * \max(\sigma_p, \rho^* \sigma_\theta)$	6.1
		m	
Mahalanobis Projection Threshold	$\lambda_{projmax}$	9.21	6.3
Projection Like Miss	$N_{likemiss}$	6	6.3
Hypothesis Maximum Time without Update	T_{max}	3 sec	6.4
Time Correlation Lateral Drift	ρ	0.9999	7
Minimum Hypothesis Probability	P_{Hmin}	0.02	3

probability of going straight and turning are similar so the MHT combines both resulting in an intermediate target position, which is out of the road centrelines. With the following measures the probability of turning to the left rapidly grows, and finally the estimated target position changes to the associated curve, as the other hypotheses get pruned. Once again the target has no lateral drift, and the lateral drift tracker is disabled.

Figs. 15 and 16 show one-run results, including lateral drift with a constant value of 3 m and enabling the lateral drift tracker: the target is not following the centre of the road. Hence the lateral drift is estimated and corrected. Specifically, Fig. 15 shows the convergence in the estimation of lateral drift ($d_{trans}(f)$), and the associated 95% error band along the central estimate, according the lateral drift Kalman filter estimator.

Fig. 16 shows the results of the whole tracking filter (MBMHF plus

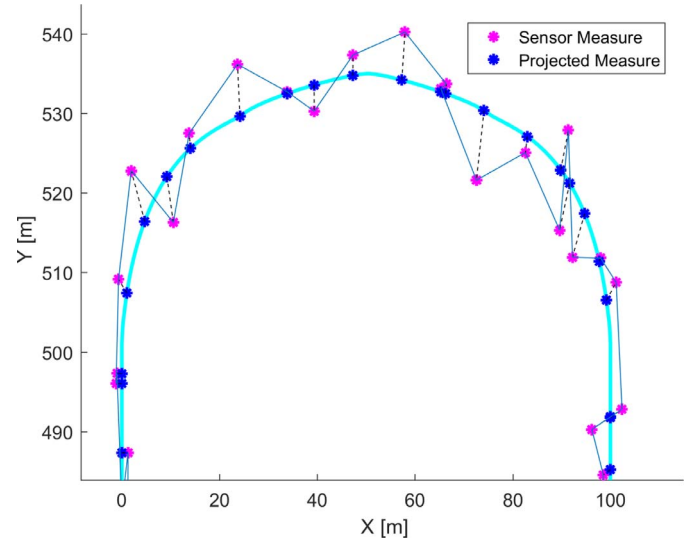


Fig. 12. Projection to a single road.

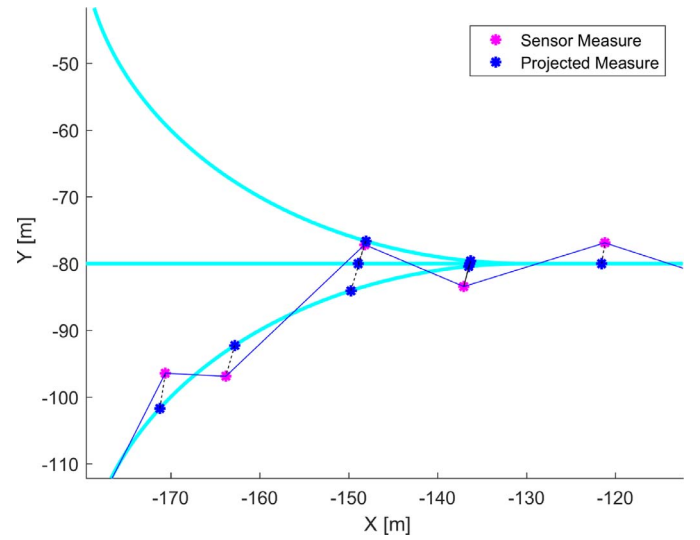


Fig. 13. Projection result in an intersection.

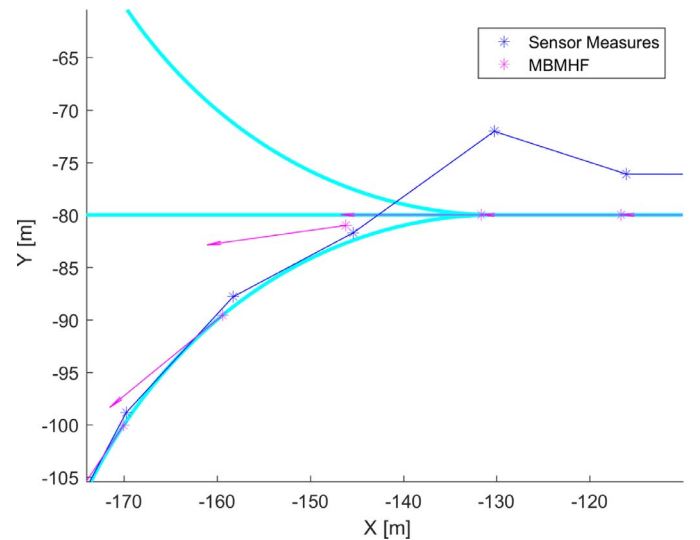


Fig. 14. Tracking result in an intersection without lateral drift.

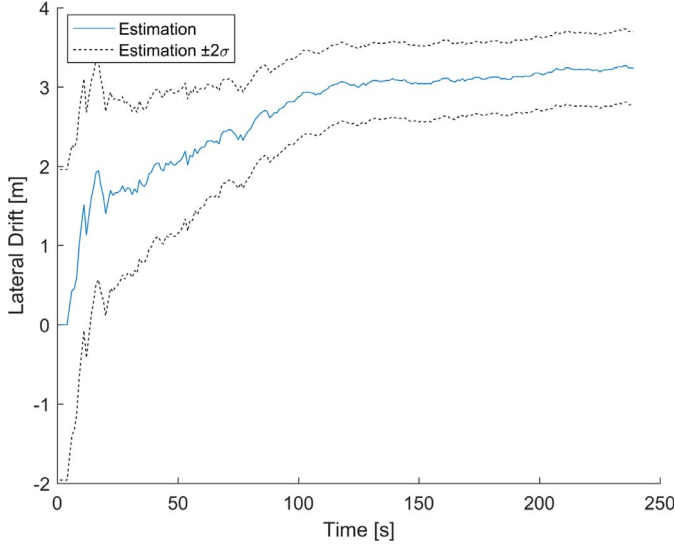


Fig. 15. Lateral drift estimation.

lateral drift tracking) in 2D, for a part of the trajectory with intersections and associated hypothesis. Here it is clear how the lateral drift is back in the output estimate and the system is robust against this drift for the localization/longitudinal tracking processes.

In order to mathematically evaluate the performance of the system we will use a Montecarlo simulation, and calculate the 95th percentile for each of the three following magnitudes:

1. Horizontal position error (euclidean distance between simulated and predicted 2D positions).
2. Absolute value of speed error.
3. Absolute value of heading error.

In order to be able to accurately estimate a safe upper bound for this percentile, with an associated confidence value of 95%, using order statistics methods [29], and a Gaussian approximation of binomial distribution, we can order the samples of the interesting magnitude in increasing order (from 1 to N) and calculate the index of the limit sample as:

$$i = N \cdot 0.95 + 1.645 \cdot \sqrt{N \cdot 0.95 \cdot 0.05} \quad (71)$$

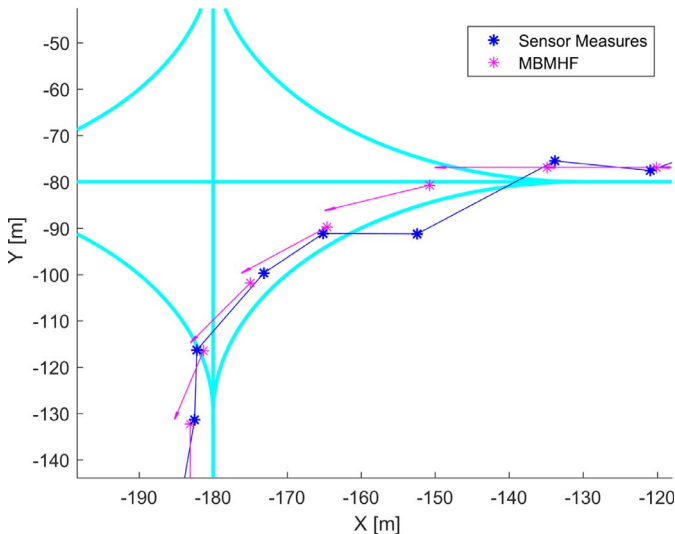


Fig. 16. Tracking result in an intersection with lateral drift.

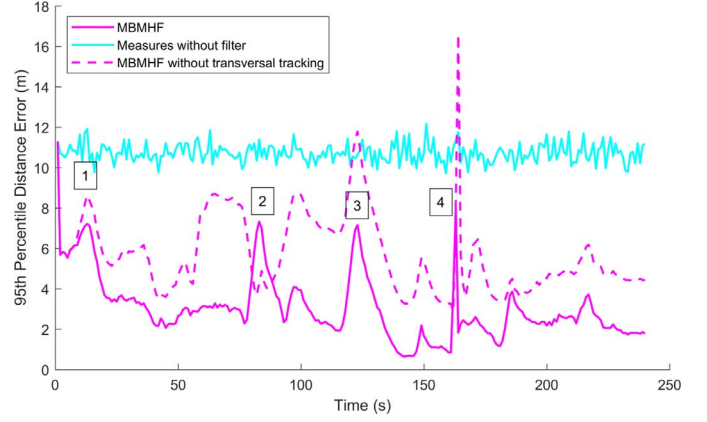


Fig. 17. Distance error.

and then select the ordered sample at the position given by the ceiling integer of i . Our tracking system was analysed by a Montecarlo simulation with 500 iterations, resulting in an i value of 483. The following figures show the 95th percentile error of the distance, groundspeed and heading errors, comparing it with the using the sensor plots without filter and with our system without transversal tracking. The ground-speed and the heading of the plots are obtained by performing the difference of each 2D Cartesian measure with the previous one, dividing by the measurement period, and translating this velocity estimate to polar coordinates.

In this scenario there are four error peaks. The first one (around second 4) is caused by the first acceleration. Second one (between seconds 77–94) occurs when the target decelerates until it's stopped. The third one (between seconds 115–142) appears when the target restart the movement. The last one (between seconds 155–170) is caused in the intersection with three options. First the results of our system compared only to the plots are presented due to clarity. The following figures show the position error, the groundspeed error and the heading error respectively (Figs. 17–19).

In order to demonstrate the quality of our tracker we compared it with several trackers in the literature, both making use of map or not. The techniques taken as reference for comparison are: a standard IMM tracker without map information (two constant velocity models with different plant noise) [7]; an efficient particle filter constrained to roads proposed in [5]; a longitudinal Kalman filter that incorporates road map information, proposed in [15]; and a IMM tracker (a circular model and an accelerated model) that utilises the map information to localise and filter, proposed in [6]. The following figures compare the results of our system with the previously cited systems. Figs. 20–22 show the 95th percentile error of the distance, groundspeed and heading respectively.

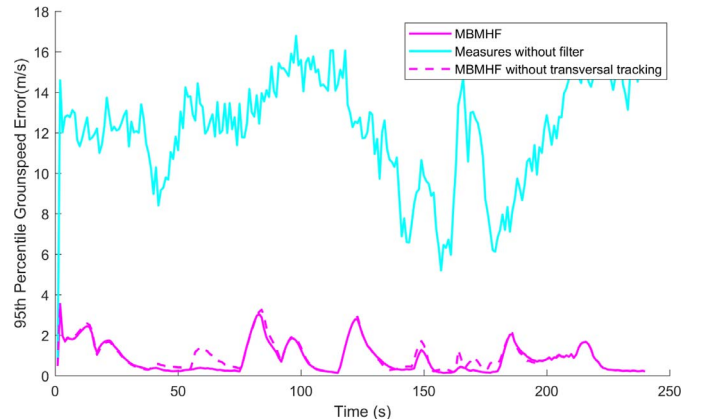


Fig. 18. Groundspeed error.

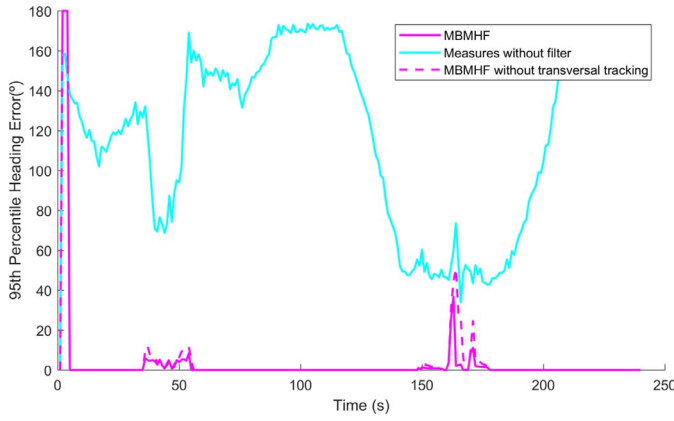


Fig. 19. Heading error.

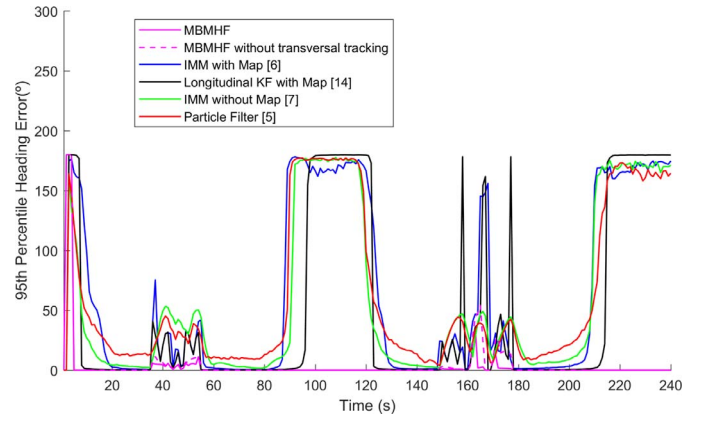


Fig. 22. Heading error for the different trackers.

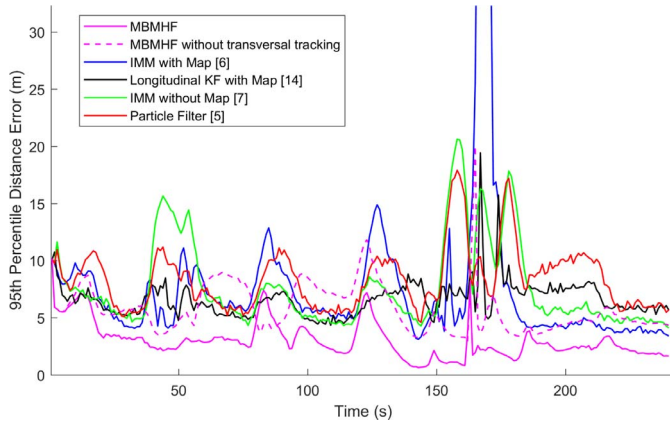


Fig. 20. Distance error for the different trackers.

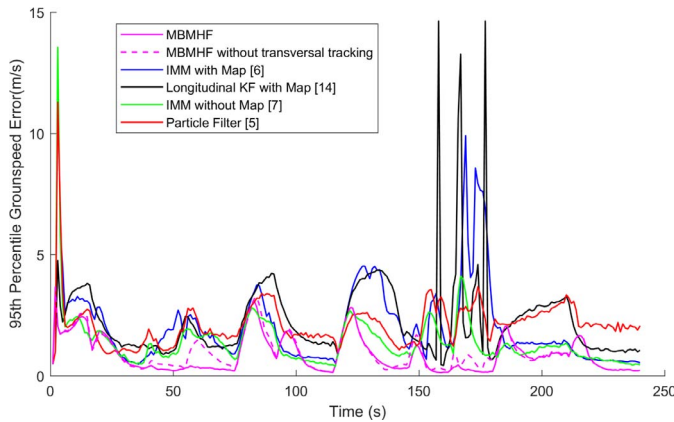


Fig. 21. Groundspeed error for the different trackers.

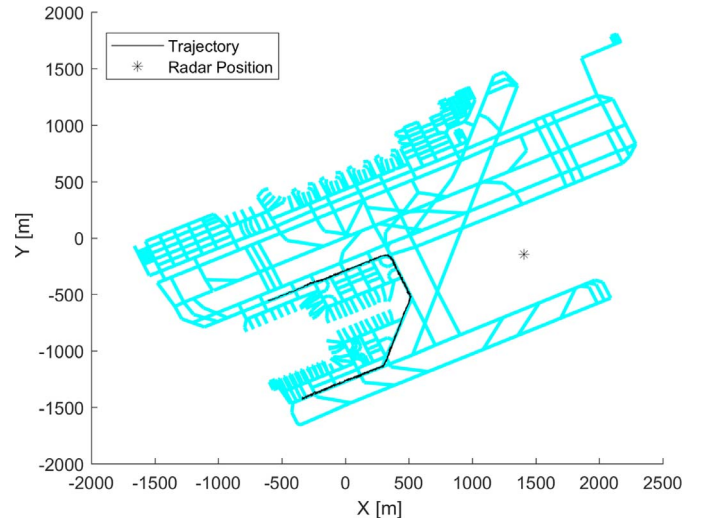


Fig. 23. Barcelona El Prat airport.

8.2. Real data results

In order to demonstrate the performance of our system with real data, even in a multisensor deployment, a scenario from the Barcelona El Prat Airport (IATA: BCN, OACI: LEBL) is also analysed. The filtered data come from an aircraft performing a taxi movement. The sensors available are a SMR and Multilateration (MLAT) system. The SMR noise covariance used in the filter is built from the standard deviations shown in Table 1 while the MLAT is assumed to have noise covariance diagonal matrix compatible with a standard deviation value of 5 m for X and Y coordinates. The airport map, the SMR position and the trajectory are shown in Fig. 23. Also, the rest of filter parameters are those in Table 3.

This application needs to track multiple targets, and the measures from the SMR and MLAT are biased, as these sensors have different measure centroids (aircraft center for SMR, transponder antenna position for MLAT, usually near the front of the aircraft). Therefore, before filtering the measures are coordinate-transformed (using stereographic projection, of wide use in Air Traffic Control), associated (using classical distance from measures to track gating and Nearest Neighbour assignment, sufficient in this application due to the reduced ratio between measurement noise and distance among targets), and then corrected using a local bias estimator, based on the methods described in [30] to ensure they are almost bias-free. All filters shared the same bias estimation and correction process, to make the comparison fair.

To check the performance of the system we compare the results

It can be concluded that both in stationary and longitudinal manoeuvres our system has better performance. It should be pointed that the heading error in the MBMHF is insignificant due to the use of the road information and even when the target stops due to the heading protection explained in Section 6.4. Another interesting conclusion is that in curves our system shows much lower error peaks, even near intersections. The IMM with map information [6] quickly degenerates in the intersections due to lack of projection in areas with multiple roads. The IMM without map [7] information presents similar performance when the target moves with constant velocity but degenerates when the target turns. The particle filter [5] reduces the manoeuvre peak of the two previous trackers but is less accurate than the proposed MBMHF.

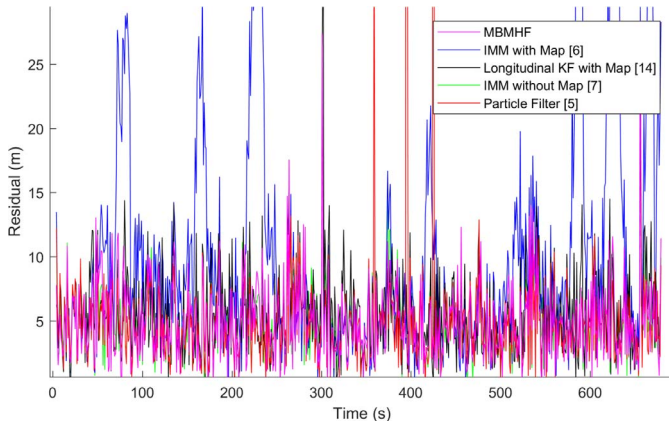


Fig. 24. Residual of the position.

Table 4
Absolute residual of the position.

Filter	Average residual [m]	95th Percentile residual [m]
MBMHF	4.89	11.21
IMM with map [6]	10.38	36.34
Longitudinal KF with map [15]	5.72	12.11
IMM without map [7]	4.92	10.93
Particle filter [5]	6.49	11.08

from the different filters. The problem now is we do not have the ideal trajectory, so we use some indirect measures of the performance.

Specifically, we use the following performance measures:

1. To assess position accuracy we measure the residual between the measures and the predicted position. This residual, ideally, has two components, one related to measurement noise (equal for all filters), and another one related to predicted filter error. So, in general, smaller residual means better predicted filter error and therefore a better filter with respect to position.
2. To assess groundspeed accuracy we measure the difference between successive filtered groundspeed values. In the actual trajectory, the groundspeed changes smoothly as the aircraft moves, but any filter will produce a somehow noisy groundspeed and may also have

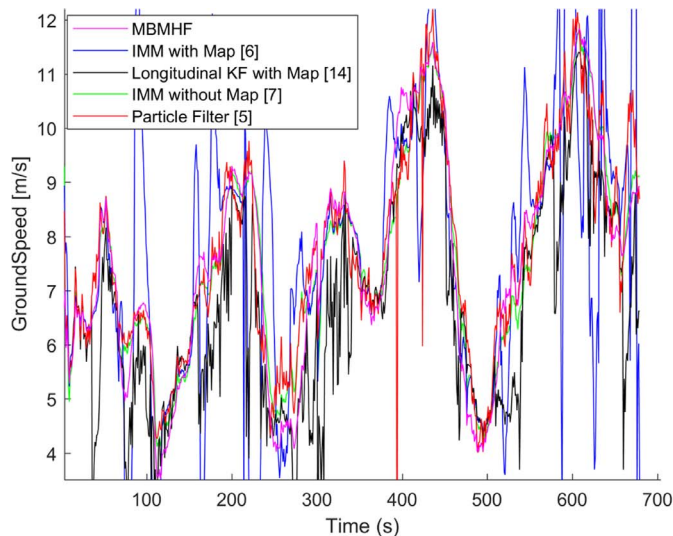


Table 5
Groundspeed absolute difference.

Filter	Aver. GS diff [m/s]	95th Percent GS diff. [m/s]
MBMHF	0.132	0.404
IMM with map [6]	0.503	2.359
Longitudinal KF with map [15]	0.454	1.963
IMM without map [7]	0.134	0.375
Particle filter [5]	0.298	0.772

rebounds near acceleration manoeuvres. So, a reduced difference between filtered groundspeed values, in general, means better filter with respect to velocity.

3. To assess heading accuracy the different between the filtered heading and the ideal heading of the segment being followed by the target is calculated. The segment can be trivially obtained offline after aircraft operation. For turning (or spline) segments the methods to obtain the ideal heading are described in Section 4.

Fig. 24 shows the absolute value of the residual of the position along time, where it can be seen that our filter has reduced error peaks than most competing algorithms and one of the smallest average values. Table 4 shows the average value of this magnitude and its 95th percentile along the trajectory. In those results the quality of the MBMHF and that of most of the competing filters are similar, but it is much better than the IMM with Map, and slightly better than the Longitudinal KF with Map.

Fig. 25 shows the estimated groundspeed for each of the filters, and the absolute difference between two consecutive groundspeed updates. It can be seen that the MBMHF and the IMM without Map have a good and stable performance while the other filters present problems in the intersections, making the groundspeed oscillate. Then, Table 5 shows the average value of the absolute difference between groundspeed updates and its 95th percentile along the trajectory. In those results the quality of the MBMHF and the IMM without Map show to be clearly better than those of the rest of the filters.

Finally, the results showing the quality of the heading estimation are depicted and analysed. In Fig. 26, the absolute heading error is depicted for all competing filters. It can be observed that the error in our system is much lower than that of all the competing filters. This is clearly seen in the Table 6, summarising along-trajectory average value of the absolute heading error and its 95th percentile.

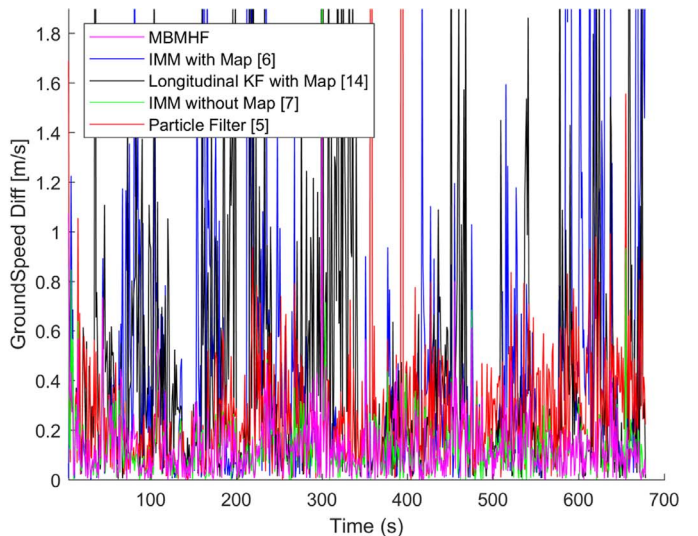


Fig. 25. Left: GroundSpeed tracking result. Right: absolute difference between groundspeed updates.

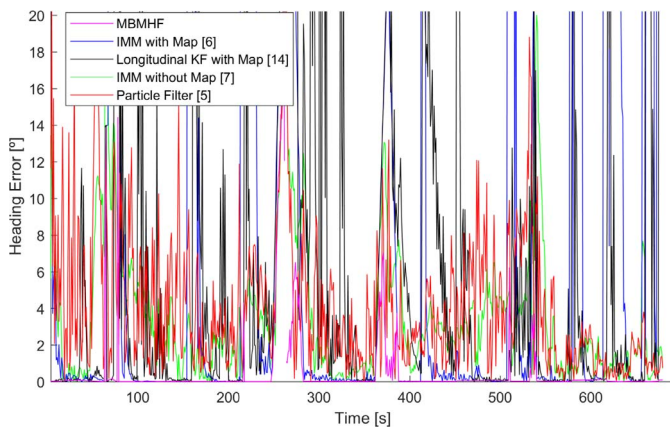


Fig. 26. Heading tracking error.

Table 6

Absolute heading error.

Filter	Aver. Abs. Heading Err. [°]	95th Abs. Heading Err. [°]
MBMHF	0.65	4.46
IMM with Map [6]	18.45	167.47
Longitudinal KF with Map [15]	6.96	42.49
IMM without Map [7]	3.60	12.70
Particle Filter [5]	4.20	13.59

Table 7

Computational load (number of track updates per second).

Filter	Track update rate (Hz)	Min. Track update rate (Hz)
MBMHF	392	80
IMM with Map [6]	7	6
Longitudinal KF with Map [15]	950	293
IMM without Map [7]	1200	290
Particle Filter [5]	16	7

8.3. Computational load results

Some results analysing the adequacy of our filter for real-time applications are shown in (Table 7). All the competing filters have been coded in a MATLAB, using a mono-thread approach, and have being executed in a computer with the an Intel i5-2500K processor (release date 2011) and 8GB-DDR3 of RAM memory. The scenario being executed is that of the real data.

We have analysed the amount of time needed to process each track update, and calculated the number of measures (plots/track updates) that the different filters could process per second. This, assuming a sensor update period of 1 s is equivalent to the number of tracks that can be updated in real time, even in this low-end computer and with not too optimised code. Also, we calculated the worst-case minimum number of measures (plots/track updates) that the different filters could process per second by doing the inverse of the time needed to process the worst-case plot in the trajectory.

It can be observed that our filter, coded in not the most efficient programming language and executed in an average computer, has a reduced computational load, only higher to that of the Longitudinal KF with Map and the IMM without Map approaches. The low performance of the particle filter is due to the amount of particles and the lack of parallel programming, while the low performance of the IMM with map is due to the amount of intersections between noise ellipses and

segments that have to be calculated. In any case, it is clear our filter is suitable for real time tracking of large amounts of targets.

9. Conclusions

This work presents a system able to track the longitudinal and transversal position of a target using road map information with high accuracy. The results show the benefits of these techniques over other proposals in the literature for ground tracking. Several key ideas are the main responsible of this benefits over other approaches:

1. The quality of the map adaptation has been improved from other approaches using only straight lines or circular arcs, by adding the use of Bézier splines to the typical segments: straight lines and arcs of circumference. This allows to represent the road map with more freedom and precision.
2. A better performance of the longitudinal tracking has been achieved by using projected measures to the road centre and a multiple hypotheses approach capable of tracking all the possible longitudinal trajectories in the intersections. The associated Bayesian hypothesis combination approach, with the quasi-orthogonal projection and longitudinal likelihood terms are quite robust and enable safe tracking of targets.
3. Optimal projection methods have been developed for all types of segments for longitudinal and lateral drift tracking, specially adequate for the cases with time-varying or space-varying measurement covariances and of non-diagonal covariance matrices.
4. The transversal drift of the target is also tracked correcting the bias between the centre of the road and the real target position.

An obvious limitation of the system is that the described approach is only valid when neither the measurement noise nor the target movement make measurements jump from a given segment area to the surroundings of another segment not connected with the previous one. As the hypothesis formation procedure just looks in connected segments, the filter would fail if this kind of jump is present. This results in limitations linking the maximum speed of the target, the measurement noise variances and the minimum length of segments. Also, limits on the minimum radius of curves, related to measurement noise variances could appear.

Future work should investigate the capability to deal with erroneous maps and the integration with out-of-road movements, and also further investigate the limitations described in the previous paragraph and find potential mitigation procedures to overcome them.

Acknowledgements

This work was supported in part by Universidad Politécnica de Madrid Project “Tecnologías Avanzadas para la Monitorización y Gestión Remota del Tráfico Aéreo de Vehículos Pilotados y no Pilotados” (RP1509550C02), by Fundación para el Desarrollo de las Telecomunicaciones (FUNDETEL) Project “Tecnologías y Sistemas de Control y Gestión de Tráfico Aéreo” (GPD.SJC.001), and by the Spanish Ministry of Economy and Competitiveness under Grant TEC2014-57022-C2-1-R.

References

- [1] T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati, I. Kadar, B. Abrams, E. Eadan, Tracking ground targets with road constraints using an IMM estimator, *Aerospace Conference*, 1998 IEEE Vol. 5, IEEE, 1998, pp. 5–12.
- [2] P.J. Shea, T. Zadra, D.M. Klammer, E. Frangione, R. Brouillard, Improved state estimation through use of roads in ground tracking, *Aerosense 2000*, International Society for Optics and Photonics, ISO 690, 2000, pp. 321–332.
- [3] E. Blasch, Z. Wang, D. Shen, H. Ling, G. Chen, Surveillance of ground vehicles for airport security, *SPIE Defense + Security*, International Society for Optics and Photonics, 2014. 90890B-90890B.
- [4] B. Pannetier, K. Benameur, V. Nimier, M. Rombaut, VS-IMM using road map

- information for a ground target tracking. *Information fusion*, 8th International Conference on Vol. 1, IEEE, 2005, pp. 8–pp. 2005.
- [5] Y. Cheng, T. Singh, Efficient particle filtering for road-constrained target tracking. *aerospace and electronic systems*, IEEE Trans. 43 (4) (2007) 1454–1469.
 - [6] J.G. Herrero, J.A.B. Portas, J.R.C. Corredra, Use of map information for tracking targets on airport surface. *aerospace and electronic systems*, IEEE Trans. 39 (2) (2003) 675–693.
 - [7] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, John Wiley & Sons, 2004.
 - [8] T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati, I. Kadar, Ground target tracking with variable structure IMM estimator. *aerospace and electronic systems*, IEEE Trans. 36 (1) (2000) 26–46.
 - [9] Y. Yang, Y. Zhao, M. Kyas, Recursive Bayesian estimation using a topological map for indoor position tracking, *Vehicular Technology Conference (VTC Spring)*, 2014 IEEE 79th, (2014), pp. 1–5. IEEE.ISO 690.
 - [10] H. Oh, S. Kim, A. Tsourdos, B. White, Road-map assisted standoff tracking of moving ground vehicle using nonlinear model predictive control, *American Control Conference (ACC)*, 2012, (2012), pp. 4263–4268. IEEE.ISO 690.
 - [11] J.S. Pyo, D.H. Shin, T.K. Sung, Development of a map matching method using the multiple hypothesis technique, *Intelligent Transportation Systems*, 2001. Proceedings. 2001 IEEE, IEEE, 2001, pp. 23–27.
 - [12] M. Tahk, J.L. Speyer, Target tracking problems subject to kinematic constraints, *IEEE Trans. Autom. Control* 35 (3) (1990) 324–326. ISO 690.
 - [13] G.A.O. Caicai, C.H.E.N. Wei, Ground moving target tracking with VS-IMM using mean shift unscented particle filter, *Chin. J. Aeronaut.* 24 (5) (2011) 622–630.
 - [14] Z. Xinyan, Z. Wei, Application of road information in ground moving target tracking, *Chin. J. Aeronaut.* 20 (6) (2007) 529–538.
 - [15] D. Streller, Road map assisted ground target tracking, *Information Fusion*, 2008 11th International Conference on, IEEE, 2008, pp. 1–7.
 - [16] C.S. Agate, K.J. Sullivan, Particle filtering algorithm for tracking multiple road-constrained targets, *Aerosense 2003*, International Society for Optics and Photonics, 2003, pp. 256–266.
 - [17] S. Gattein, B. Pannetier, P. Vannoorenberghe, Analysis and integration of road projection methods for multiple ground target tracking, 2005 7th International Conference on Information Fusion, volume 1, IEEE, 2005, pp. 8–pp.
 - [18] B. Pannetier, K. Benameur, V. Nimier, M. Rombaut, Ground moving target tracking with road constraint, *Defense and Security*, International Society for Optics and Photonics, 2004, pp. 138–149.
 - [19] L. Xu, X.R. Li, Z. Duan, J. Lan, Modeling and state estimation for dynamic systems with linear equality constraints, *IEEE Trans. Signal Process.* 61 (11) (2013) 2927–2939.
 - [20] M. Ulmke, W. Koch, Road-map assisted ground moving target tracking. *aerospace and electronic systems*, IEEE Trans. 42 (4) (2006) 1264–1274.
 - [21] L. Cuiping, S. Jinping, M. Shiyi, L. Desheng, Tracking ground targets with road constraint using multiple hypotheses tracking, *Signal Processing Systems (ICSPS)*, 2010 2nd International Conference on, volume 2, IEEE, 2010, pp. V2–265.
 - [22] G. Thomaidis, M. Tsogas, P. Lytrivis, G. Karaseitanidis, A. Amditis, Multiple hypothesis tracking for data association in vehicular networks, *Inf. Fusion* 14 (4) (2013) 374–383.
 - [23] J.G. Herrero, J.B. Portas, F.J. Rodriguez, J.C. Corredra, Surface movement radar data processing methods for airport surveillance, *IEEE Trans. Aerosp. Electron. Syst.* 37 (2) (2001) 563–585. ISO 690.
 - [24] C. Hasberg, S. Hensel, M. Westenkirchner, K. Bach, Integrating spline curves in road constraint object tracking, *Intelligent Transportation Systems*, 2008. ITSC 2008. 11th International IEEE Conference on, IEEE, 2008, pp. 1009–1014.
 - [25] H. Prautzsch, W. Boehm, M. Paluszny, *Bzier and B-spline Techniques*, Springer Science & Business Media, 2013. ISO 690.
 - [26] K. Kastella, M. Biscuso, Tracking algorithms for air traffic control applications, *Air Traffic Control Q.* 3 (1) (1995) 19–43. ISO 690.
 - [27] S.S. Blackman, *Multiple-Target Tracking with Radar Applications*, 463 Artech House, Inc., Dedham, MA, 1986, p. 1. 1986.
 - [28] Y. Bar-Shalom, P.K. Willett, X. Tian, *Tracking and Data Fusion*, YBS publishing, 2011.
 - [29] L.J. Bain, M. Engelhardt, *Introduction to Probability and Mathematical Statistics*, Brooks/Cole. ISO 690, 1987.
 - [30] J.A. Besada, G. De Miguel, A.M. Bernardos, J.R. Casar, Generic multisensor multi-target bias estimation architecture, *IET Radar Sonar Navig.* 6 (5) (2012) 365–378.