Along-track Localization for Cooperative Autonomous Vehicles

Elwan Héry[1], Philippe Xu[1] and Philippe Bonnifait[1]

Abstract—Localization is a key problem for autonomous vehicle navigation. The use of high-definition maps and perception algorithms allows now to have lane-level accurate pose estimation in terms of cross-track and heading error. In this paper, we focus on the along-track localization of cooperative vehicles. We introduce a one-dimensional formulation of the localization problem by considering curvilinear coordinates. The covariance intersection filter is derived in one dimension leading to a minimum variable filter which allows multiple vehicles to cooperate while keeping consistent localization estimates. We show that the along-track localization error is directly dependent on the relative orientation between the trajectories followed by the cooperating vehicles. Experiments with two autonomous electric vehicles were conducted to evaluate the proposed approach.

Fig. 1. Curvilinear pose $[s \; n \; \psi]^T$ of the mobile frame along curve $\mathcal{C}$ from origin p_1. H is the orthogonal projection of point M on $\mathcal{C}$.

I. Introduction

Vehicle localization is a well-known problem and has been studied for decades [9]. Nevertheless, computing lane-level accurate and consistent estimates of the vehicle pose, i.e., position and heading, is still an open issue and remains a key feature to reach fully autonomous driving. In the past few years, the availability of high-definition maps [1], [3], which gather lane-level information such as lane markings, has led to new map-aided localization algorithms [8]. At the same time, the development of vehicle-to-vehicle (V2V) wireless communication devices allows a vehicle to have more sources of information through the exchange of messages.

When high-definition maps are available, it is possible to use exteroceptive sensors such as cameras to estimate the vehicle pose with respect to the map, e.g., by doing map-matching with lane markings [15]. Using lane markings information is especially useful to reduce cross-track and heading errors. However, in terms of along-track localization, lane markings can be invariant, for example on straight roads, therefore making the localization more challenging along the vehicle longitudinal direction [8]. Through out this paper, we assume that the vehicle has the capability to properly estimate its cross-track location and heading with respect to a given lane. We focus the work on along-track localization by neglecting cross-track and heading estimation errors.

Cooperative localization using V2V communications enables a vehicle to use information coming from the sensors of its surrounding vehicles to refine its own state estimation [6], [11], [13]. In the context of decentralized localization,
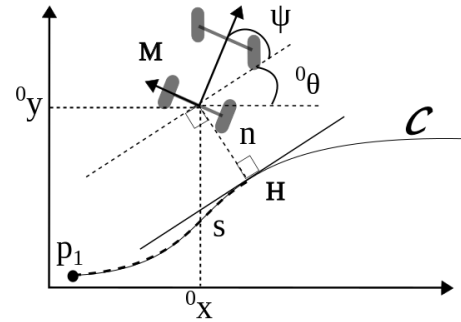
i.e., each vehicle has its own estimates of the state of world, one main issue is to handle the data incest problem where the exchanged pieces of information may be highly correlated. The covariance intersection filter [10] is one of the most popular approach to tackle the fusion of non-independent pieces of information. While being a pessimistic combination rule, it is guaranteed to provide consistent estimates [5], [14], which is in particular critical for autonomous driving.

In this paper, we introduce a novel formalization of vehicle localization by focusing only on along-track localization. We assume that a vehicle is capable of estimating its cross-track location and its heading as well as estimating the relative pose of the vehicles in its surrounding. The main idea resulting from such assumptions is to be able to handle the localization uncertainty in a one-dimensional space, more precisely, in the curvilinear space attached to the road [4], [7]. In a one-dimensional formulation, we show that the covariance intersection algorithm is very easy to use and is equivalent to a minimum variance operator.

In Sec. II, we first introduce the problem of map-aided along-track localization and the one-dimensional covariance intersection filter in the context of cooperative vehicles. In Sec. III, we provide a detailed derivation of the geometrical transformations to compute curvilinear coordinates. Then, in Sec. IV, we present the fusion strategy for cooperative localization. Finally, in Sec. V, some experimental results are provided and discussed.

II. Along-track cooperative localization

A. Along-track localization

In the context of autonomous vehicles, the pose estimation of the vehicle is often computed with respect to a planned trajectory. This trajectory is itself geometrically attached to

[1]The authors are with Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, 60203 Compiègne Cedex, France `firstname.surname@hds.utc.fr`

a map, where the center of the current lane can be seen as the ideal path. In this paper, localizing a vehicle with respect to the map is equivalent to localizing it w.r.t. its planned trajectory. The pose estimation error of a vehicle with respect to the map can therefore be decomposed in three terms: along-track, cross-track and heading errors. These three error terms are easily represented in curvilinear coordinates.

Let $\mathcal{C}$ be a curve in a two-dimensional Euclidean space and $^0q_c = \begin{bmatrix} ^0x & ^0y & ^0\theta \end{bmatrix}^T$ a pose in the global Cartesian frame R_0. We define point H as the orthogonal projection of point $\begin{bmatrix} ^0x & ^0y \end{bmatrix}^T$ on $\mathcal{C}$, i.e., the point on the curve $\mathcal{C}$ minimizing its distance to ^0M. We define the curvilinear pose $^0q_s = \begin{bmatrix} s & n & \psi \end{bmatrix}^T$ where s is the length of the curve from its origin point up to point H, n is the signed Euclidean distance between the two points ^0M and H and ψ is the relative angle between $^0\theta$ and the tangent line of the curve at point H as illustrated in Fig. 1.

The along-track localization problem is then formulated as estimating the curvilinear abscissa s of a pose while assuming that both n and ψ are known. The estimation problem is therefore only considered in one dimension.

B. Covariance intersection filter

Let $\mathbf{X_1}$ and $\mathbf{X_2}$ be two multivariate random variables representing estimates of the system true state with associated covariance matrices $\mathbf{\Sigma_1}$ and $\mathbf{\Sigma_2}$. In a classical Kalman filter approach, the two estimates are combined as follows:

$$\mathbf{\Sigma}^{-1} = \mathbf{\Sigma_1}^{-1} + \mathbf{\Sigma_2}^{-1}, \tag{1}$$
$$\mathbf{X} = \mathbf{\Sigma}(\mathbf{\Sigma_1}^{-1}\mathbf{X_1} + \mathbf{\Sigma_2}^{-1}\mathbf{X_2}). \tag{2}$$

The above equations are based on the assumption that X_1 and X_2 are independent. In the case where this assumption is not verified, Julier and Uhlmann [5] proposed to use the covariance intersection fusion which leads to the following estimates:

$$\mathbf{\Sigma}^{-1} = \omega\mathbf{\Sigma_1}^{-1} + (1-\omega)\mathbf{\Sigma_2}^{-1}, \tag{3}$$
$$\mathbf{X} = \mathbf{\Sigma}(\omega\mathbf{\Sigma_1}^{-1}\mathbf{X_1} + (1-\omega)\mathbf{\Sigma_2}^{-1}\mathbf{X_2}), \tag{4}$$

where $\omega \in [0,1]$ is chosen so as to minimize the determinant or the trace of $\mathbf{\Sigma}$.

In the one-dimensional case with estimates x_1 and x_2 and variances σ_1^2 and σ_2^2, the covariance intersection algorithm can be rewritten as

$$\sigma^2 = (\omega/\sigma_1^2 + (1-\omega)/\sigma_2^2)^{-1}, \tag{5}$$
$$x = \sigma^2(\omega x_1/\sigma_1^2 + (1-\omega)x_2/\sigma_2^2). \tag{6}$$

Minimizing the variance σ^2 in (5) is equivalent to maximizing:

$$\omega(1/\sigma_1^2 - 1/\sigma_2^2) + 1/\sigma_2^2, \tag{7}$$

which is the equation of a line. If $\sigma_1^2 < \sigma_2^2$, then the maximum is reached for $\omega = 1$ which leads to $\sigma^2 = \sigma_1^2$ and $x = x_1$. On the contrary, if $\sigma_1^2 > \sigma_2^2$, then we have $\sigma^2 = \sigma_2^2$ and $x = x_2$. One can see that the covariance intersection filter in one dimension is equivalent to keeping the estimate with the smallest variance.

III. CURVILINEAR COORDINATES

A. Polyline map

We define a map as a set of geolocalized points $\mathcal{M} = \left\{ p_i = [x_i \ y_i]^T, \ i \in \{1,\ldots,n\} \right\}$ defining a curve represented as a polyline, i.e., a sequence of adjacent line segments. In practice, this curve corresponds to the center of the road.

In order to make the coordinate transformation less computationally demanding, a Frenet frame for each origin point p_i can be computed offline as a transformation matrix $^iT_0$ defined as follows:

$$^iT_0 = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i & 0 & -x_i\cos\alpha_i - y_i\sin\alpha_i \\ -\sin\alpha_i & \cos\alpha_i & 0 & x_i\sin\alpha_i - y_i\cos\alpha_i \\ 0 & 0 & 1 & -\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

where $\alpha_i = \text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i)$ for $i = 1,\ldots,n-1$. The transformation matrix $^iT_0$ is based on a homogeneous matrix, the third column and the third row are added to take the heading of the vehicle into account in the transformation. An enhanced map $\mathcal{M}^* = \left\{ (p_i, {}^iT_0), \ i \in \{1,\ldots,n-1\} \right\}$ can then been defined.

Therefore, with this formalism, one can compute directly the pose in the i-th Frenet frame R_i:

$$\begin{bmatrix} ^iq_c & 1 \end{bmatrix}^T = {}^iT_0 \cdot \begin{bmatrix} ^0q_c & 1 \end{bmatrix}^T. \tag{9}$$

The inverse transformation matrix is defined as:

$$^0T_i = {}^iT_0^{-1} = \begin{bmatrix} \cos\alpha_i & -\sin\alpha_i & 0 & x_i \\ \sin\alpha_i & \cos\alpha_i & 0 & y_i \\ 0 & 0 & 1 & \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

B. Curvilinear coordinates transformation

The first step to compute the curvilinear coordinates of a pose ^0q_c with respect to a polyline is to find the line segment closest to this point. To do so, the pose is first transformed into all the Frenet frames ^iq_c attached to each line segment using the transformation matrices $^iT_0$. The closest line segment can be chosen as the one for which the norm $\sqrt{^ix^2 + {}^iy^2}$ is the smallest for instance, if there is little ambiguity in the map-matching process. In this work, the polyline is considered as the reference path to follow which is actually a sub-part of the global map but with no junctions.

Once the closest line segment has been computed, two cases can occur. In the nominal case, the Frenet abscissa ^ix is greater than zero and it is kept as it is (see Fig. 2-a). Conversely, if $^ix < 0$ then the Frenet abscissa is set to zero, point M in Fig. 2-b is projected on point H. The complete procedure is detailed in Algorithm 1 in which the function distPointSegment returns the distance MH as displayed in Fig. 2. In the first case (Fig. 2-a), ^iq_s is equal to a Cartesian pose in the i-th Frenet frame ^iq_c. Otherwise (Fig. 2-b),

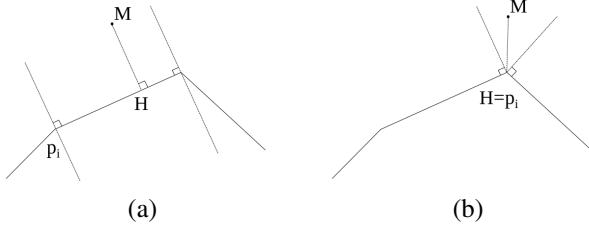$$^iq_s = \begin{bmatrix} 0 & sign\left(^iy\right)\|MH\| & ^i\psi \end{bmatrix}. \tag{11}$$

Fig. 2. Distances from points to segments. In case (a), point M is projected orthogonaly onto the point H. In case (b), all the points between the two dotted lines, such as point M, are projected onto the same point H.

Algorithm 1 Transformation of a Cartesian pose into a curvilinear pose.

1: ^0q_s **= global2Curv**$\left(^0q_c, \mathcal{M}^*\right)$
2: $i = \underset{i \in \{0,\dots,n-1\}}{\operatorname{argmin}} \left(\text{distPointSegment}\left(^0q_c, [p_i p_{i+1}]\right)\right)$
3: $^0q_s = {}^iq_s + \left[\ \sum_{j=0}^{i-1} \|p_{j+1} - p_j\|\ \ 0\ \ 0\ \right]^T$

Once ^iq_s as been computed, the global curvilinear abscissa of the pose ^0q_s accumulates the length of all the preceding segments.

The inverse transformation from curvilinear coordinates to global coordinates is detailed in Algorithm 2. The principle consists to find the index of the segment that is one step ahead and to subtract the cumulative length of every segments.

Algorithm 2 Transformation of a curvilinear pose into a Cartesian pose.

1: ^0q_c **= curv2Global**$\left(^0q_s, \mathcal{M}^*\right)$
2: **for** $i = 2, \dots, n-1$
3: **if** $s < \sum_{j=1}^{i-1} \|p_{j+1} - p_j\|$
4: $\begin{bmatrix} ^0q_c \\ 1 \end{bmatrix} = {}^0T_i \begin{bmatrix} s - \sum_{j=1}^{i-2} \|p_{j+1} - p_j\| \\ n \\ \psi \\ 1 \end{bmatrix}$
5:   **break**
6: **end if**
7: **end for**
8: **end**

C. Uncertainty propagation

In this article, the error of a curvilinear pose estimate $^0q_s = \begin{bmatrix} s & n & \psi \end{bmatrix}^T$ is only considered along the curvilinear abscissa direction. Therefore the covariance matrix associated to ^0q_s can be written as

$$\Sigma_s = \begin{bmatrix} \sigma_s^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (12)$$

In order to transform the curvilinear pose to the global Cartesian frame ^0q_c by using the function **curv2Global**$\left(^0q_s, \mathcal{M}^*\right)$, we first need to remove the cumulative segment lengths in
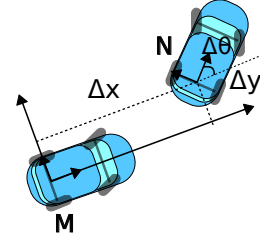

Fig. 3. The relative pose of vehicle N w.r.t. vehicle M is represented as $^M\Delta T_N = [\Delta x\ \Delta y\ \Delta \theta]^T$.

the curvilinear abscissa which does not change the variance, then use the transformation matrix $^iT_0$ where i is the index of the line segment to which ^0q_s is matched. This linear transformation leads to the following covariance matrix

$$\Sigma_c = \sigma_s^2 \begin{bmatrix} \cos^2 \alpha_i & \cos \alpha_i \sin \alpha_i & 0 \\ \cos \alpha_i \sin \alpha_i & \sin^2 \alpha_i & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (13)$$

Using the same reasoning, a pose in the global frame $^0q_c = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ with covariance matrix

$$\Sigma_c = \begin{bmatrix} \sigma_x^2 & \sigma_{x,y} & \sigma_{x,\theta} \\ \sigma_{x,y} & \sigma_y^2 & \sigma_{y,\theta} \\ \sigma_{x,\theta} & \sigma_{y,\theta} & \sigma_\theta^2 \end{bmatrix}, \quad (14)$$

can be transformed into a curvilinear pose ^0q_s **= global2Curv**$\left(^0q_c, \mathcal{M}^*\right)$ with covariance matrix

$$\Sigma_s = \begin{bmatrix} \sigma_s^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (15)$$

where $\sigma_s^2 = {}^0\sigma_x^2 \cos^2 \alpha_i + {}^0\sigma_y^2 \sin^2 \alpha_i + 2 {}^0\sigma_{x,y} \cos \alpha_i \sin \alpha_i$.

A vehicle M can use V2V communication to receive the global pose estimate of another vehicle N. If vehicle M can also detect the other vehicle, it can compute the relative pose of vehicle N in its local frame. With these two pieces of information, vehicle M can have another estimate of its own global pose. The relative pose $^M\Delta T_N = \begin{bmatrix} \Delta x & \Delta y & \Delta \theta \end{bmatrix}^T$ is illustrated in Fig. 3. In the global frame, the pose estimate $^0q_c^N = \begin{bmatrix} x^N & y^N & \theta^N \end{bmatrix}^T$ of a vehicle N can be transformed into a pose estimate $^0q_c^M = \begin{bmatrix} x^M & y^M & \theta^M \end{bmatrix}^T$ of a vehicle M as follows:

$$\begin{aligned} ^0q_c^M &= {}^0q_c^N - \begin{bmatrix} cos(\theta^M) & -sin(\theta^M) & 0 \\ sin(\theta^M) & cos(\theta^M) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^M\Delta T_N \\ &= \begin{bmatrix} x^N - \Delta x \cos(\theta^M) + \Delta y \sin(\theta^M) \\ y^N - \Delta x \sin(\theta^M) - \Delta y \cos(\theta^M) \\ \theta^N - \Delta \theta \end{bmatrix} \quad (16) \end{aligned}$$

In this paper, it is assumed that the values of Δx, Δy, $\Delta \theta$ and θ^N are known with certainty, i.e., we have a perfect sensor for relative pose estimation (e.g., LiDAR or camera) then the pose estimates $^0q_c^M$ and $^0q_c^N$ share the same covariance matrix.

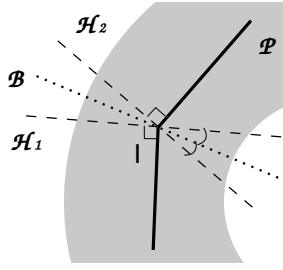Fig. 4. Discontinuity near an apex. The zone between the two orthogonal lines $\mathcal{H}_1$ and $\mathcal{H}_2$ introduces a discontinuity in the curvilinear frame.

Finally, by propagating the covariance matrix of the curvilinear pose estimate ${}^0q_s^N$ to ${}^0q_s^M$, using (13) and (15), we show that

$$\sigma_s^{M\,2} = \sigma_s^{N\,2} \cos^2 \Delta\alpha, \tag{17}$$

where $\Delta\alpha = \alpha_j - \alpha_i$ is the relative angle between the two line segments i and j to which the vehicles M and N are matched to respectively. We can see that when the two vehicles are driving along two parallel trajectories, the curvilinear position uncertainty is not reduced. On the contrary, if the two vehicles are on perpendicular trajectories, then they are both perfectly localized. In the rest of the paper, we assume that vehicle M represents the host vehicle, exponent ^{M} will be omitted for clarity.

D. Discontinuity issues

The use of a polyline is simple because the data of the map present it self as points. However, the polyline is not spatially differentiable. It introduces some problems of continuity in terms of heading and along-track localization near the apexes. Indeed, when the position of the vehicle is between the two orthogonal lines $\mathcal{H}_1$ and $\mathcal{H}_2$ (see Fig. 4) the orientation jumps from an orientation relative to the first segment of P to an orientation relative to the second segment of P.

The second problem is on the curvilinear abscissa computation. When the vehicle is between the orthogonal lines $\mathcal{H}_1$ and $\mathcal{H}_2$ on the left side of the polyline (in the case depicted in Fig. 4), the curvilinear abscissa remains the same, it is the curvilinear distance between the beginning of the polyline and the point I. When the vehicle is between the orthogonal lines but at the right side of the polyline, the curvilinear abscissa jumps from one point of one segment of P to a point of the other segment. This discontinuity happens when the vehicle cross the bisecting line $\mathcal{B}$. Therefore, the change from the global frame to the curvilinear frame is not bijective.

In our practical case, this problem of discontinuity is not severe, due to the discretized position localization and the small angle between adjacent segments, it is quite unlikely to have several consecutive position estimates falling near the apexes of the polyline.

One way to overcome this issue is to generate a differentiable curve going through the points of the map [12], [16] or to use a non-Euclidean projection space using lanelets

---

Algorithm 3 One-dimensional cooperative Kalman filter.

1: **Prediction**
2: $\quad \hat{s}_{k|k-1} = \hat{s}_{k-1|k-1} + T_e u_k \cos(\psi_{k-1})$
3: $\quad \hat{\sigma}_{s,k|k-1}^2 = \hat{\sigma}_{s,k-1|k-1}^2 + Q_{k-1}$
4: **GNSS update**
5: $\quad K_k = \hat{\sigma}_{s,k|k-1}^2 / (\hat{\sigma}_{s,k|k-1}^2 + \sigma_{z_s,k}^2)$
6: $\quad \hat{s}_{k|k} = \hat{s}_{k|k-1} + K_k(z_{s,k} - \hat{s}_{k|k-1})$
7: $\quad \hat{\sigma}_{s,k|k}^2 = \hat{\sigma}_{s,k|k-1}^2 (1 - K_k)^2 + \sigma_{z_s,k}^2 K_k^2$
8: **Covariance intersection**
9: **if** $\hat{\sigma}_{s,k|k}^2 > \sigma_{z_s^N,k}^2 \cos^2 \Delta\alpha$
10: $\quad \hat{s}_{k|k} = z_{s,k}^N$
11: $\quad \hat{\sigma}_{s,k|k}^2 = \sigma_{z_s^N,k}^2 \cos^2 \Delta\alpha$
12: **end if**

---

as presented in [1]. The drawback of these two approaches is that the curvilinear transformations become highly non linear, therefore making the propagation of uncertainty much more challenging.

IV. Cooperative along-track Kalman filter

In this paper, we implemented a one-dimensional Kalman filter using the curvilinear abscissa as state vector. In Algorithm 3, we provide the different steps of the Kalman filter. At the prediction stage, the previous estimate $\hat{s}_{k-1|k-1}$ is used with the current control input u_k (longitudinal velocity) with a time sampling T_e to predict the current state. Then at the update stage, the curvilinear abscissa estimate coming from GNSS measurements $z_{s,k}$ is used to update the current estimate. Finally, the covariance intersection filter is used to add another update step using the curvilinear abscissa computed from another vehicle $z_{s,k}^N$. Fig. 5 provides a detailed flow chart of the whole system including the geometrical transformation.

V. Experimental results

Experiments were conducted using two autonomous electric cars (see Fig. 6) following each other on the Seville experimental road (see Fig. 7). The road is composed of two roundabouts linked by a straight road. The vehicles were equipped with an INS/IMU Novatel's SpanCPT with RTK corrections in order to compute the ground truth poses. Low cost Ublox 8T GNSS receivers were used to provide localization data. The wheel speed from the vehicle CAN bus was also recorded. The relative pose was computed by using the true poses from the SpanCPTs. The same one-dimensional cooperative localization algorithm was implemented in both vehicles.

We compare the covariance intersection approach with a naive Kalman filter in which the covariance intersection step is replaced by another Kalman update which assumes that the pose estimate received from the other vehicle is independent of its own estimate.

We first considered a simulated scenario where the global poses of both vehicles were generated by adding a white noise with a standard deviation of 1m to the ground truth
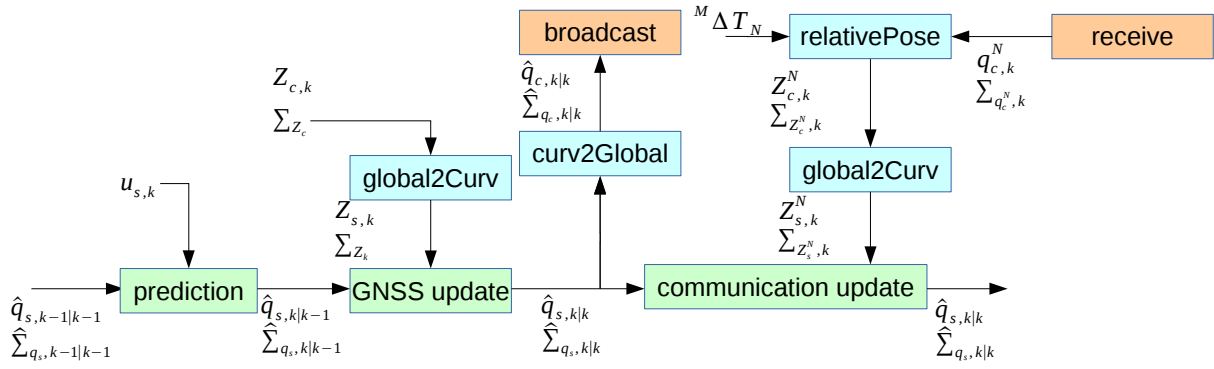
Fig. 5.   Data fusion flow chart.

Fig. 6.   ApAChe autonomous electric cars. The white vehicle is leading the way while the gray one is platooning.
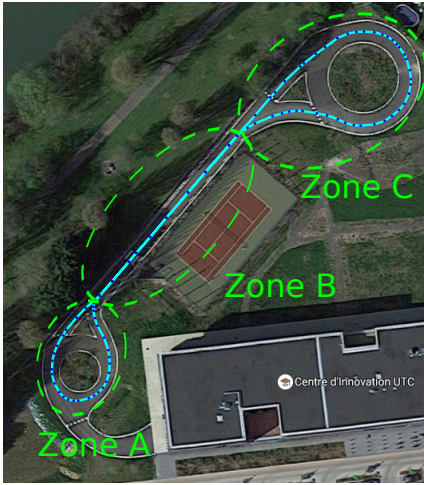

Fig. 7.   Seville experimental road (Google Maps). The road is composed of two roundabouts (zones A and C) linked by a straight road (zone B).

TABLE I

OUT-OF-BOUND RATE AND MEAN SQUARE ERROR FROM SIMULATION.

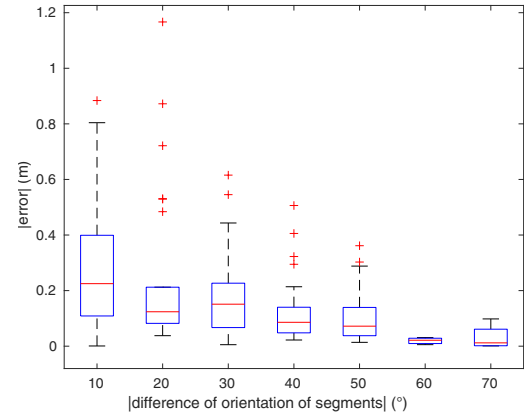|	Follower		Leader	
	τ (%)	$\bar{e}$ (m)	τ (%)	$\bar{e}$ (m)
	$\sigma = 1$m		$\sigma = 1$m	
KF without communication	5	0.41	5	0.31
KF with communication	9.71	0.20	10.29	0.20
CI with communication	3.82	0.25	2.35	0.25
	$\sigma = 1$m		$\sigma = 0.01$m	
KF without communication	5	0.41	5	0.077
KF with communication	5.88	0.071	6.18	0.071
CI with communication	5.88	0.074	5	0.077

Fig. 8.   Box plot of the |error| w.r.t. relative angle between the segments matched by the two vehicles (CI with communication).

poses. One can see from Tab. I that the Kalman filter using communication is not consistent. Indeed, the percentage of out-of-bound estimates τ reaches 10%. The covariance intersection remains consistent although being quite pessimistic ($\tau < 5\%$). The mean squared errors $\bar{e}$ for the three methods indicate that the accuracy is increased when using communication. In a second scenario, the white noise added to the leading vehicle was reduced to 0.01m. Because the covariance intersection filter is equivalent to a minimum variance filter, we can see that the high localization accuracy

of the leader is transmitted to the following vehicle.

Fig. 8 displays the error distribution w.r.t. to different relative angle between the two line segments to which the two vehicles are matched. One can verify that when the vehicles are sharing information the localization error decreases with the angle between the segments matched by the two vehicles, as expected by Eq. (17).

In Tab. II, we show the results using real localization data from Ublox 8T GNSS receivers. One can see that the over-confidence of the Kalman filter is even more significant in this situation, leading to an out-of-bound rate of more than

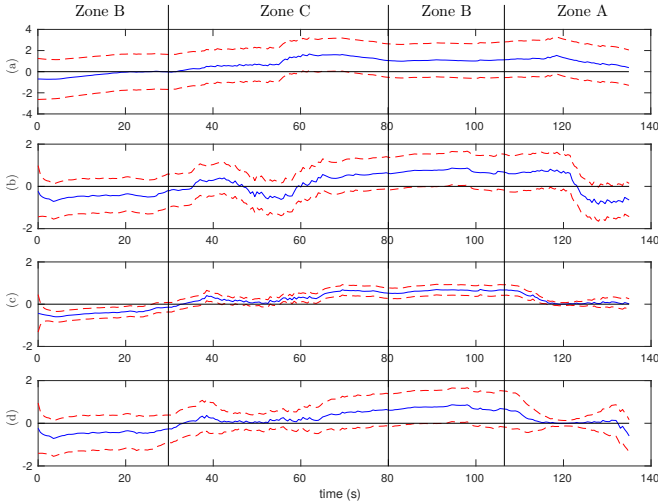|	Follower		Leader	
	τ (%)	$\bar{e}$ (m)	τ (%)	$\bar{e}$ (m)
KF without communication	5.17	0.55	5.90	0.96
KF with communication	61.62	0.44	69.00	0.43
CI with communication	4.80	0.46	4.43	0.45

Fig. 9. Along-track errors with 95% confidence intervals (red dashed).
(a) and (b) represent the localization errors of the leading and following
vehicles, respectively, using a Kalman filter without any communication. (c)
shows the errors of the leader using a Kalman filter with communication.
(d) shows the errors of the leader using the covariance intersection filter.

60%. This is explained by the fact that the errors of the state
estimates of the two vehicles are already non independent
even before exchanging information. Fig. 9 (a) and Fig. 9 (b)
represent the localization errors of the leading and following
vehicles, respectively, using a Kalman filter without any
communication. One can see that the localization errors of
the two vehicles are correlated. Fig. 9 (c) shows the errors
of the leader using a Kalman filter with communication. One
can clearly see the non-consistency of the Kalman filter. By
using the covariance intersection filter, one can see from Fig.
9 (d) that the estimates remain consistent.

In Fig. 9, one can also see that the confidence intervals
are narrower around 40-60s (zone C) and around 120-140s
(zone A). This is exactly in line with the theory as it was
when the two vehicles where in the roundabouts that is when
the difference between the two orientations was large.

VI. CONCLUSION

This paper has formalized a one-dimensional localization
problem for cooperative localization of autonomous vehicles
by introducing curvilinear abscissa along a given reference
path extracted from a map. We showed that multiple vehicles
can cooperate to reduce their along-track localization error
while remaining consistent by using the covariance intersec-
tion filter which is equivalent to a minimum variance filter
in one dimension. It has been shown that classical Kalman

filter data fusion is sensitive to the data incest problem. It was
also shown that the error reduction is directly linked to the
relative orientation of one vehicle with respect to the other.
This makes our algorithm particularly efficient for curved
roads and intersections, since the confidence interval can be
significantly reduced in this case. The results show also that,
in a context of a platoon of vehicles, a single well localized
vehicle is sufficient to localize the whole platoon.

REFERENCES

[1] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient map representa-
tion for autonomous driving. In *IEEE Intelligent Vehicles Symposium
Proceedings*, pages 420–425, June 2014.
[2] L. C. Bento, Ph. Bonnifait, and U. J. Nunes. Cooperative GNSS
positioning aided by road-features measurements. *Transportation
Research Part C: Emerging Technologies*, 79:42–57, 2017.
[3] D. Bétaille and R. Toledo-Moreo. Creating enhanced maps for lane-
level vehicle navigation. *IEEE Transactions on Intelligent Transporta-
tion Systems*, 11(4):786–798, Dec 2010.
[4] K. Chu, M. Lee, and M. Sunwoo. Local path planning for off-
road autonomous driving with avoidance of static obstacles. *IEEE
Transactions on Intelligent Transportation Systems*, 13(4):1599–1616,
December 2012.
[5] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in
the presence of unknown correlations. In *Proceedings of the American
Control Conference*, volume 4, pages 2369–2373, June 1997.
[6] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis. Localization of
a group of communicating vehicles by state exchanged. In *IEEE/RSJ
International Conference on Intelligent Robots and Systems*, pages
519–524, October 2006.
[7] J. Kim, K. Jo, W. Lim, M. Lee, and M. Sunwoo. Curvilinear-
coordinate-based object and situation assessment for highly automated
vehicles. *IEEE Transactions on Intelligent Transportation Systems*,
16(3):1559–1575, June 2015.
[8] F. Kuhnt, S. Orf, S. Klemm, and J. M. Zöllner. Lane-precise localiza-
tion of intelligent vehicles using the surrounding object constellation.
In *IEEE 19th International Conference on Intelligent Transportation
Systems*, pages 526–533, 2016.
[9] J. Levinson and S. Thrun. Robust vehicle localization in urban envi-
ronments using probabilistic maps. In *IEEE International Conference
on Robotics and Automation*, pages 4372–4378, 2010.
[10] H. Li and F. Nashashibi. Cooperative multi-vehicle localization
using split covariance intersection filter. In *IEEE Intelligent Vehicles
Symposium*, pages 211–216, June 2012.
[11] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using
relative observations. In *IEEE International Conference on Robotics
and Automation*, pages 2797–2802, April 2005.
[12] D. S. Meek and D. J. Walton. Planar G2 Hermite interpolation with
some fair, C-shaped curves. *Journal of Computational and Applied
Mathematics*, 139(1):141–161, February 2002.
[13] F. de Ponte Müller, E. M. Diaz, B. Kloiber, and T. Strang. Bayesian
cooperative relative vehicle positioning using pseudorange differences.
In *IEEE/ION Position, Location and Navigation Symposium*, pages
434–444, May 2014.
[14] M. Reinhardt, B. Noack, and U. D. Hanebeck. Closed-form opti-
mization of covariance intersection for low-dimensional matrices. In
International Conference on Information Fusion, pages 1891–1896,
July 2012.
[15] Z. Tao, Ph. Bonnifait, V. Frémont, J. Ibanez Gusman, and S. Bonnet.
Road-centred map-aided localization for driverless cars using single-
frequency GNSS receivers. *Journal of Field Robotics*, June 2017.
[16] H. Wang, J. Kearney, and K. Atkinson. Arc-length parameterized
spline curves for real-time simulation. In *Proc. 5th International
Conference on Curves and Surfaces*, pages 387–396, 2002.