



Map-based Curvilinear Coordinates for Autonomous Vehicles

Elwan Héry, Stefano Masi, Philippe Xu, Philippe Bonnifait

► To cite this version:

Elwan Héry, Stefano Masi, Philippe Xu, Philippe Bonnifait. Map-based Curvilinear Coordinates for Autonomous Vehicles. 20th IEEE International Conference on Intelligent Transportation Systems (ITSC 2017), Oct 2017, Yokohama, Japan. pp.1-7. hal-01596909v3

HAL Id: hal-01596909

<https://hal.science/hal-01596909v3>

Submitted on 18 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Map-based Curvilinear Coordinates for Autonomous Vehicles

Elwan Héry, Stefano Masi, Philippe Xu and Philippe Bonnifait

Sorbonne universités, Université de Technologie de Compiègne, CNRS UMR 7253 Heudiasyc

57 Av. Landshut CS 60319, 60203 Compiègne cedex, France

{elwan.hery - stefano.masi - philippe.xu - philippe.bonnifait}@hds.utc.fr

Abstract—Localization within high definition maps is a key problem for autonomous navigation as vehicles need to extract information from them. In addition, many navigation tasks are defined with respect to map features. For instance, estimating the cross-track and along-track gaps of a vehicle with respect to a given path is critical for lane keeping or intersection management. Map-based localization is also important for cooperative tasks like platooning in curved roads or lane changing. This work studies different methods to compute map-based coordinates defined as curvilinear abscissa, lateral distance and heading with respect to paths in high definition maps. Four approaches using polylines, lanelets and splines are compared. Thanks to real experiments, the discontinuity issues of polylines used in current high definition maps are evaluated and we discuss advantages and drawbacks of splines-based and lanelet methods. We also report experimental results corresponding to a platoon of two vehicles in curved roads and evaluate the effects of the use of low cost GNSS receivers.

I. INTRODUCTION

Localization is an important problem for autonomous vehicles in particular for cooperative navigation using wireless information exchange. Many works have been made, but accurate lane level localization remains an open issue.

A Cartesian frame is often used as it facilitates the manipulation of geometric transformations. For vehicle control, localization has often to be done with respect to a path, usually attached to a lane [6]. To achieve autonomous lane keeping for instance, a vehicle needs to know at what distance it is from the center of the lane, this distance being used for lateral control. Autonomous vehicles need also to know their position along the lanes, i.e., curvilinear abscissa along the center path. This information can be used for longitudinal control, e.g., to stop at an accurate position or to drive in platoon. In this paper, we are interested in using curvilinear pose [4], [7] defined as a curvilinear abscissa, a signed lateral distance and a relative orientation with respect to a reference path describing the center of the lane.

Different methods can be used to obtain a curvilinear pose from a Cartesian pose. They depend on the road and lane models. A simple polyline [5] composed of a sequence of segments is often used because of its easy implementation in real-time. However, discontinuity issues may appear if the path is not sufficiently discretized. The lanelet model also uses polylines as geometric representation but provides continuous tangents at the junctions of the segments thanks to non-Euclidean point-to-segment projection [1]. This approach

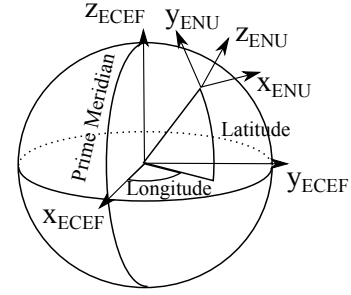


Figure 1: Geodetic, ECEF and ENU coordinates systems.

provides an interesting real-time solution without discontinuity issues. Other models using continuous curves like splines can also be used to avoid discontinuities [3], [8], [9] even if they are computationally demanding and more difficult to implement in real-time embedded systems [11]. In this paper, these different approaches are studied and compared. In Sec. II, we first present different coordinates systems used for localization and discuss the advantages of using a coordinates system more adapted to navigation using the map as reference. Then, in Sec. III, we present the map used for the experiments and the different studied models: polylines, lanelets and splines. Finally, in Sec. IV, we discuss the results regarding discontinuity and map-matching issues for different road models.

II. COORDINATES SYSTEMS FOR AUTONOMOUS NAVIGATION

In navigation, different coordinates systems can be used for different purposes. Geodetic coordinates are classically used to localize vehicles with GNSS receivers. Latitude and longitude are two angles from the center of the Earth which characterize the 2D position of the vehicles, the height being given with respect to an ellipsoidal model of the Earth. This reference is often the WGS 84 geodetic system since it is used by the GPS system. Since many geometric transformations are complicated, and sometimes need approximations, with this coordinates system, a Cartesian coordinates system is more convenient.

The ECEF (Earth-Center Earth-Fixed) is a Cartesian coordinates system centered at the center of the Earth. This Cartesian coordinates system is not well adapted for vehicles

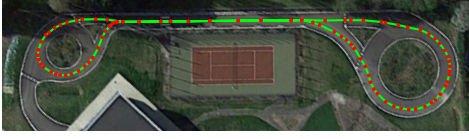


Figure 2: Map of the experimental road called Seville (the bird view image is from Google Map). The center of the lanes is depicted in green and represented as a set of red points.

moving on the surface of the Earth: the center is too far from the navigation zone and the axes are not oriented with the horizontal plan where simplifications can be made to work with 2D poses.

The ENU (East North Up) coordinates system is a Cartesian coordinates system lying on the surface of the Earth. The three axes are aligned with the East, North and vertical directions. The East and North axes are tangent to the ellipsoid model of the Earth.

For autonomous vehicles navigation, another coordinates system is very useful. Indeed, for the control of a vehicle, knowing where it is in a Cartesian frame may not be representative of the driving state of the vehicle. For automatic control purposes, it is most interesting to know where the vehicle is with respect to the road. In order to use localization information directly, curvilinear coordinates can be used instead. The abscissa corresponds to the curvilinear abscissa along the center of the lane, the ordinate is a signed lateral distance from the center of the lane and the heading is the relative orientation with respect to the lane. The lateral distance and the relative orientation with respect to the center of the lane can be used for lateral control directly. The curvilinear abscissa is well adapted for longitudinal control. For example, with communicating vehicles sending their positions and receiving the positions of others, the difference between two curvilinear abscissas gives directly the correct inter-distance between two vehicles even when the road is curved and even if the two vehicles do not detect each other with perception sensors such as cameras or LiDARs.

III. ROAD MODELS

In the same way as geographic coordinates, a reference model is needed for curvilinear coordinates. Throughout this paper, we assume that an autonomous vehicle is following a path which corresponds to the center of a set of lanes extracted from a map. Our aim is to provide a road model suitable for localization and autonomous navigation. During the experiments, we used a high definition map of the test road displayed on Fig. 2. This map contains many attributes such as the centers of the lanes and the lane markings. The map is composed of roads of multiple lanes. These lanes are discretized as polylines. A path is a concatenation of several following lanes.

A. General curvilinear coordinates

Let us define the curvilinear coordinates $[s n \psi]^T$, as illustrated in Fig. 3, with the curvilinear abscissa s (useful for

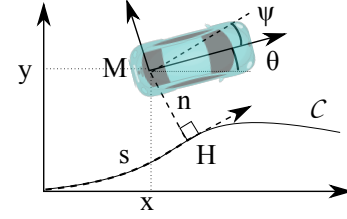


Figure 3: Curvilinear pose coordinates $[s n \psi]^T$ relative to curve C (curvilinear abscissa s , signed lateral distance n and relative orientation ψ) and Cartesian pose $[x y \theta]^T$ of the mobile frame attached to point M the center of the body frame of the vehicle.

longitudinal control), the lateral distance n from the reference map and the angle ψ between the orientation of the vehicle (heading θ) and the angle of the tangent of the reference map. To compute a curvilinear pose from a Cartesian pose, the first step is to find point H of the map with the smallest distance to point M . Then, the curvilinear abscissa can be computed. With the use of the tangent at point H , the signed lateral distance n and the relative angle ψ can be derived to compute the curvilinear pose. n is negative when M is at the right of the tangent and positive when M is at the left. To find a Cartesian pose from a curvilinear pose, point H is found by using the curvilinear abscissa s , and after having computed the tangent at point H , the signed lateral distance n gives the Cartesian point M . This tangent also gives the Cartesian orientation of the vehicle by using the relative orientation ψ .

B. Polyline model

The map is given by a sequence of points located on the center of the lanes defined as follow: $\{p_i = [x_i y_i]^T, i \in \{1, \dots, n\}\}$. A simple choice to model the road is to use a polyline per lane, i.e., line segments between each consecutive points.

To compute curvilinear coordinates, one first needs to do a map-matching which is to search for the smallest distance among the distances from point M to each segments of the polyline. In case of a complex network or when the estimate is inaccurate, this basic map-matching procedure has to be robustified using heading information for instance.

The distance from a point to a segment is illustrated in Fig. 4. When the point is between the orthogonal line of the starting point of the segment \mathcal{L}_i and the orthogonal line of the ending point \mathcal{L}'_i like in Fig.4 (a), the distance is the orthogonal distance. When this point is between the orthogonal line of the end point of the previous segment \mathcal{L}'_{i-1} and the orthogonal line of the starting point \mathcal{L}_i , which is at the same point p_i , as illustrated in Fig. 4 (b), the distance is the one from point M to this common point p_i . The segment with the shortest distance is the map-matched segment. Point H associated to this map-matching is either the orthogonal projection in case (a) or the starting point p_i in case (b). This point is used to compute the curvilinear abscissa which is the sum of the length of each segment from the first one of the road to the map-matched one (not included) and the distance from the

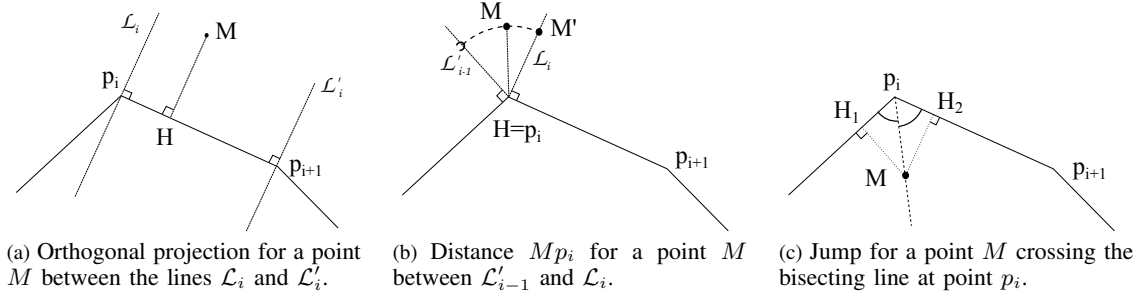


Figure 4: Polyline map-matching.

starting point p_i to H . The curvilinear ordinate is the shortest distance used for the map-matching. The curvilinear ordinate is negative when point M is on the right side of the oriented polyline. The curvilinear orientation is the difference between the orientation in the global frame and the orientation of the map-matched segment.

To find a Cartesian pose from a curvilinear pose, it is possible to find the point on the polyline with the curvilinear abscissa and then to use the lateral distance to find the Cartesian position. With the tangent at the point of the polyline, the orientation in the Cartesian frame can be derived. However, one can see that in the case (b), the transformation is not bijective. A choice must be made. It can be point M' or any other point among all feasible points. In all cases, a discontinuity appears.

This method is easy to computed in real-time. Unfortunately, the polyline is not differentiable at the junctions between two points. This discontinuity of the tangent leads to discontinuity issues near these joint-points. In case (b), the arc length remains the same and every point M of the zone between the lines L'_{i-1} and L_i is projected like point M' . In case (c), two projections appear and the shortest distance depends on where point M is with respect to the bisecting line. When this line is crossed, a jump appears in the curvilinear abscissa. When the map-matched segment changes, a jump appears also on the curvilinear orientation, which is relative to this segment.

C. Spline model

One way to remove the discontinuity problem is to use a smooth curve instead of a polyline. Each segment can be replaced by vectors of two polynomial functions representing a curve to obtain at least a C^1 continuity. In this paper, we only consider polynomials up to degree 3 which correspond to cubic splines.

The spline can be defined as follows:

$$\left\{ p_i(t) = \begin{pmatrix} p_{i,x}(t) \\ p_{i,y}(t) \end{pmatrix}, i \in \{0, \dots, n\}, t \in [0; 1] \right\}, \quad (1)$$

where $p_{i,x}(t)$ and $p_{i,y}(t)$ are two polynomials parameterized by t . Every point of the curve corresponds to a $p_i(t)$. The tangent at point $p_i(t)$ equals its derivative $p'_i(t)$ and the curvature equals the second order derivative $p''_i(t)$.

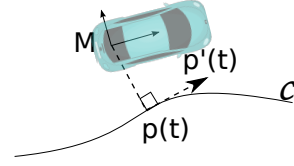


Figure 5: Map-matching of point M with spline C defined by the vector of polynomial $p(t)$.

To find the closest point of the spline from point $M = [x \ y]^T$, we have to find the roots of a polynomial. All the minimum and maximum can be found when $(M - p_i(t)) \perp p'_i(t)$, so we solve the equation

$$(M - p_i(t)) \cdot p'_i(t) = 0. \quad (2)$$

To obtain the result faster, we start by searching the closest vector of polynomials corresponding to an index i_{min} , then we only have to find the roots for this single index. To find this index i_{min} , the distance between M and every joint point $p_i = p_i(0)$ has to be computed. When the minimal distance is found for an index i , the index is $i_{min} = i$ if the projection of M in the tangent at the joint point $p'_i = p'_i(0)$ is positive and $i_{min} = i - 1$ otherwise.

The scalar product is done between $M - p_i(t)$, a vector of polynomials of degree 3, and $p'_i(t)$, a vector of polynomials of degree 2, therefore leading to a vector of polynomials of degree 5. There exists no general closed-form solution to find the roots of a polynomial of degree 5. However, numerical approaches exist to solve this problem. Overall, there exists at most five roots, the complex roots and the ones not in $[0, 1]$ can be eliminated. We select at the end the parameter t_{min} so that $\|M - p_{i_{min}}(t_{min})\|$ is the smallest distance. This distance is the curvilinear ordinate n , a minus sign is added when point M is on the right of the tangent $p'_{i_{min}}(t_{min})$.

To compute the arc length s , we need to integrate $\|p'_{i_{min}}(t)\|$ as follows:

$$s = \int_0^{t_{min}} \|p'_{i_{min}}(t)\| dt + \sum_{j=0}^{i_{min}-1} \left(\int_0^1 \|p'_j(t)\| dt \right). \quad (3)$$

The norm of a polynomial of degree 2 is the square root of a polynomial of degree 4. As there is no closed-form solution for Eq. (3), the integration must be done numerically. In order to make the computation faster, one can compute in advance the arc length associated to each p_j . Then, only the first term of Eq. (3) needs to be computed on-the-fly.

To find a Cartesian pose from a curvilinear pose, we need to solve the inverse of Eq. (3), that is to compute t_{\min} given the curvilinear abscissa s . Again, numerical approaches need to be used to solve this problem. An alternative solution is to approximate the inverse and the arc-length by a polynomial or another function [11].

Two different cubic splines are studied in this paper. The B-Spline which is an approximation of the map but which gives a smooth trajectory to follow and the Hermite spline which is an interpolation of the map and so a better representation of this map. This last spline is nevertheless less comfortable if the control points are not well positioned.

1) *B-Spline model*: A B-Spline is an approximation of the path which is not passing through the points of the path $\{p_i, i \in \{1, \dots, n\}\}$. One can compute the polynomials of a B-Spline as follows [3]:

$$\begin{pmatrix} p_{i,x}(t) \\ p_{i,y}(t) \end{pmatrix} = \begin{pmatrix} p_{i-1} & p_i & p_{i+1} & p_{i+2} \end{pmatrix} \cdots \quad (4)$$

$$\times \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}.$$

The advantages of using an approximation is to have a smoother curve more comfortable for control. However, this approximation can be problematic in some cases, e.g., the spline may cut through the corner at T-junctions.

2) *Hermite spline model*: In the case where one needs a curve going through all the points of the polyline, an interpolation can be used instead. This can be done using a Hermite spline with polynomials defined as follows [8]:

$$\begin{pmatrix} p_{i,x}(t) \\ p_{i,y}(t) \end{pmatrix} = \begin{pmatrix} p_i & t_i & p_{i+1} & t_{i+1} \end{pmatrix} \cdots \quad (5)$$

$$\times \begin{pmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix},$$

where $\{t_i, i \in \{1, \dots, n\}\}$ are the tangents of these control points, they can be computed as $t_i = (p_{i+1} - p_{i-1})/2$.

D. Lanelet model

Another strategy to tackle the problem of discontinuities in the neighborhood of the junction points is to use a map-matching algorithm based on a non-Euclidean metric as shown in Fig. 6. This conducts to the computation of a new frame, called lanelet frame [1], where the matched point on the

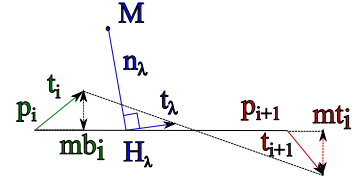


Figure 6: Lanelet map-matching of point M on a segment.

polyline segment is obtained as a convex combination of the starting and ending points of the given segment:

$$p_\lambda = \lambda p_i + (1 - \lambda) p_{i+1}. \quad (6)$$

In order to compute properly such a reference frame, one uses the constraint that n_λ and t_λ , respectively the abscissa and the ordinate axis in the new frame, must be orthogonal, i.e. $n_\lambda \cdot t_\lambda = 0$. From this constraint, one can compute λ as follows:

$$\lambda = \frac{x + y \cdot mb_i}{l - y(mt_i - mb_i)}, \quad (7)$$

where mb_i and mt_i are the ordinates of the tangent vectors in the form $t_i = (1, mb_i)$ and $t_{i+1} = (1, mt_i)$ in the local frame of the i_{th} segment. Once the origin has been computed, we compute the new frame abscissa axis called t_λ again as a convex combination of two vectors called respectively t_i and t_{i+1} which have been previously computed for every polyline segment start and end points:

$$t_\lambda = \lambda t_i + (1 - \lambda) t_{i+1}. \quad (8)$$

Finally, considering the normal vector n_λ , orthogonal to t_λ , computed as $n_\lambda = M - p_\lambda$, one can obtain the distance from point M to the i_{th} polyline segment considering $\|n_\lambda\|$ as a criterion for choosing the polyline segment for the matching.

It can be shown that with this method, under the constraint that the point that needs to be matched is close enough to the polyline segments, there are no discontinuities and we can have also a partition of the space (i.e., a set of regions in which all the points will be matched to a certain polyline segment) by only computing the normal vector for every tangent vector in every junction point of the structure.

The conversion from Cartesian coordinates to curvilinear ones and vice versa is very easy in this context. Once one knows the index i of the matched polyline segment: $s = \sum_{k=1}^{i-1} l_k + \lambda l_i$, where l_k is the length of the k_{th} polyline segment. n is the second component of n_λ in the local frame, the first component being null.

Vice versa, if we have a couple (s, n) and we want to come back to the Cartesian coordinates, we compute the corresponding λ as follows:

$$\lambda = \frac{s - \sum_{k=1}^{i-1} l_k}{l_i}, \quad (9)$$

where the index i is found by subtracting iteratively from s the length of the k_{th} segment of the polylines, until it becomes negative or null.

Given λ and the index i of the matching polyline, it is easy to compute again H_λ and t_λ , and with the knowledge of t_λ , we compute the coordinates of the starting point in the local frame $M_l = R \cdot t_\lambda / \|t_\lambda\| \cdot n$, where $R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is the rotation matrix computed for a rotation angle of $\frac{\pi}{2}$ to obtain the normal of the normalized tangent vector $t_\lambda / \|t_\lambda\|$. After this computation, it remains to translate point M_l into the global frame. It is important to underline that once the criterion for matching the polylines is fixed, this method is bijective and easy to compute.

IV. RESULTS

In this section, we compare the different road models presented before. A synthetic case study is first presented in order to illustrate discontinuity and non-linearity issues. Then, real data recorded on the experimental road Seville are used to illustrate a platooning scenario. We also compare the along-track and cross-track localization difference using the four different models.

A. Discontinuity and non-linearity issues

Let us first consider a synthetic case study where a vehicle is driving at a constant velocity along a straight line. In Fig. 7 (a), the dashed line depicts the vehicle trajectory. The goal is to compute the curvilinear coordinates of the vehicle with respect to a path which is represented by four control points, illustrated by the red circles in Fig. 7 (a). If we use the polyline or the lanelet model, the path is represented by drawing line segments between the control points (red line). For spline-based model, the curve goes through all the control points (green line) for the Hermite spline while it is not the case for the B-spline (blue line).

1) *Discontinuity and non-linearity*: In Fig. 7 (b), one can see that the curvilinear abscissa evolves differently for the four road models. First of all, for the polyline model (red curve), one can clearly see that there is a stationary point when reaching the second control point at around one meter. The curvilinear abscissa remains constant while the vehicle is still moving along the x -axis. Then, when reaching the third control point, there is a discontinuity of the curvilinear abscissa which shows a jump. For the lanelet road-model (black curve), there is no discontinuity of the curvilinear abscissa. However, contrary to the polyline model, the curve between two control points is not linear. This means that even though the vehicle is moving at constant velocity, the derivative of the curvilinear abscissa within the lanelet frame is not constant. Moreover, the derivative of the curvilinear abscissa is not continuous at the control points, which means that infinite acceleration appears when reaching a control point. The same phenomenon appears for the polyline models.

The continuity issues of the curvilinear abscissa and its derivative can be avoided by using the spline-based road models. One can see that the curvilinear abscissa curves are continuous and differentiable for both the Hermite spline (blue curve) and the B-spline (green curve). A drawback of these

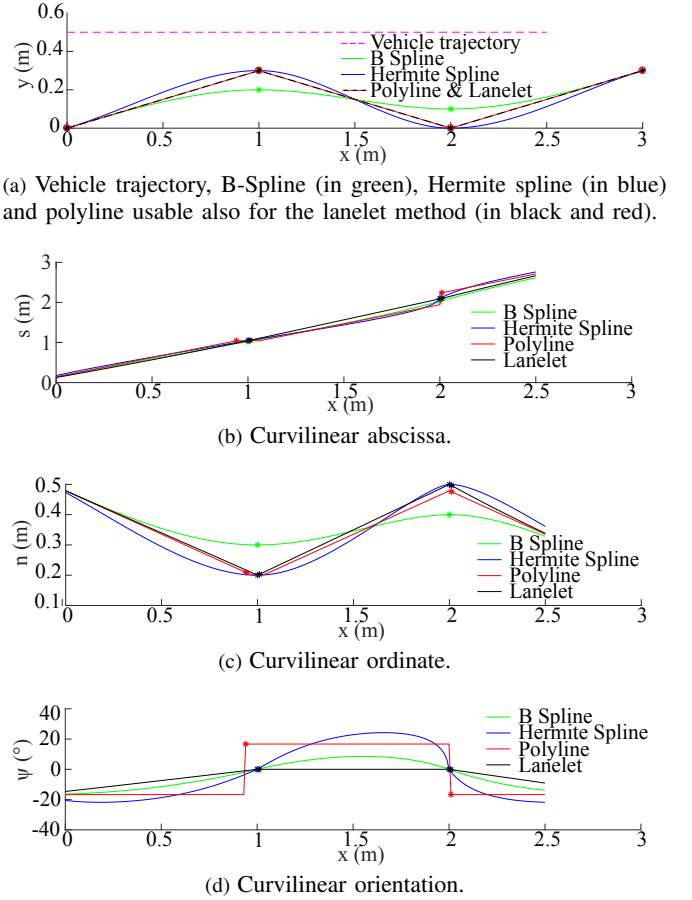


Figure 7: Discontinuity issues on the curvilinear pose of a vehicle using B-Spline (in green), Hermite spline (in blue), polyline (in red) or lanelets (in black).

two approaches is that the non-linearity is even more important than in the case of the lanelet. It is particularly obvious for the Hermite spline.

In Fig. 7 (c), one can see that the curvilinear ordinate is always continuous for the four methods. Nevertheless, the splines stay smoother and, at the joint points, a break appears in the curvilinear ordinate computed with the polyline and the lanelet. The derivative of the curvilinear ordinate of these two methods are also not continuous at the joint points.

In Fig. 7 (d), one can see that the curvilinear orientation computed with the polyline jumps when the matched segment changes. The curvilinear orientation of the lanelet method is continuous but breaks appear at the joint points and, like for the curvilinear ordinate, the derivative is not continuous. The splines methods stay always smooth with at least a C^2 continuity even at the joint points.

The different properties of the road models are summarized in Table I.

2) *Map-matching discontinuity*: In the previous case study, we showed that the curvilinear coordinates are continuous for the lanelet and spline-based models. However, the results are

	Polyline	Lanelet	Splines
s continuity		✓	✓
\dot{s} continuity			✓
s linearity	✓		
n continuity	✓	✓	✓
\dot{n} continuity			✓
n linearity	✓		
ψ continuity		✓	✓
$\dot{\psi}$ continuity			✓
ψ linearity	✓		
real-time	✓	✓	

Table I: Properties of the different road models.

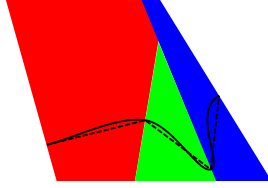


Figure 8: Map-matching discontinuity. The red, green and blue zones correspond to geographic spaces for which the map-matching results to the first, second and third lane segments, respectively.

actually valid only if the map-matching process is continuous itself, i.e., the vehicle is map-matched to each lane segments consecutively one after another. In cases where the vehicle is far away from the center of the lanes, the map-matching process may become discontinuous which will, inevitably, lead to non-continuous curvilinear coordinates.

Fig. 8 shows the map-matched zones to each segment for the lanelet and splines methods. Because the tangents at the control points are the same for these three methods, the limits between the different zones are the same. One can see that if a vehicle drives on the upper part with a distance long enough to the path, it will go from the red zone directly to the blue zone without going through the segment represented by the green zone. This will lead to a non-negligible discontinuity in the curvilinear coordinates.

Nevertheless, within the context of an autonomous vehicle, it is reasonable to assume that the vehicle always remains at a fair distance w.r.t. the center of the lanes. In such case, curvilinear coordinates using lanelet or spline road-models can remain continuous.

B. Platooning using curvilinear abscissa

As stated in Sec. 2, curvilinear abscissa can be used for platoons of cooperative vehicles. In complex scenarios, such as crossing a roundabout, using perception devices like a camera or a LiDAR will fail in providing a relevant distance from one vehicle to another. Typically, if two vehicles are at opposite poses in a roundabout, perception sensors return a distance equal to the diameter while the difference using the curvilinear abscissa returns the semi-circumference. By using curvilinear abscissa, the shape of the road is directly taken into account for platooning.



Figure 9: Map (blue) and trajectory of the vehicles (red).

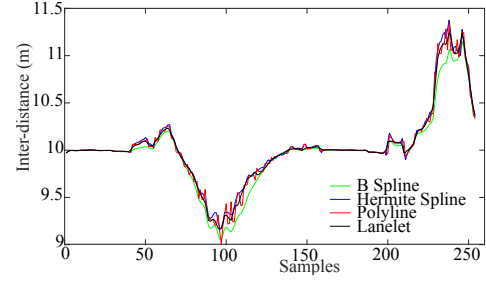


Figure 10: Curvilinear abscissa difference between two vehicles driving with a gap of ten meters.

In order to illustrate a platooning case scenario, we recorded the trajectory of an experimental Renault Zoé vehicle driving on the experimental road Seville and equipped with a NovAtel SPAN-CPT GNSS/IMU receiver with RTK corrections to provide accurate localization data. The recorded trajectory has been resampled and interpolated to simulate a constant velocity driving. Then, the same trajectory has been duplicated and time-shifted in order to simulate the platooning of two vehicles driving at the same velocity and with a constant distance of ten meters between them.

Fig. 10 shows the difference in curvilinear abscissa between the two platooning vehicles. In theory, because the two vehicles were following exactly the same trajectory and were driven at a constant velocity, the curvilinear distance between them should have remained constant and equals to ten meters. This would have been the case if the vehicles were driving exactly on the center of the lanes. In practice, it was not the case and one can see in Fig. 9 that in the bottom roundabout the vehicle drove slightly inside the reference trajectory while it was the opposite for the other roundabout. Because of this, the curvilinear abscissa became longer in one of the roundabout (samples around 40 to 140) and shorter in the other (samples between 200 and 250). In terms of vehicle longitudinal control, if the aim was to maintain a constant distance between the two vehicles, the following vehicle would have accelerated in one of the roundabout and decelerated in the other one.

For the polyline method (red curve), one can clearly see many discontinuities which may lead to a very uncomfortable longitudinal control. These discontinuities disappear for the three other models. It is interesting to note that the results from the lanelet model (black curve) are between the ones from the Hermite spline (green curve) and B-spline (blue curve) models. Indeed, the B-spline is an approximation of the polyline and is shorter than this polyline, while the Hermite spline is an interpolation and is longer than the polyline. These three methods would lead to a more comfortable longitudinal

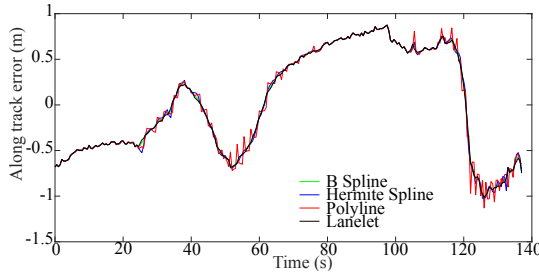


Figure 11: Along-track errors with a low cost GNSS receiver.

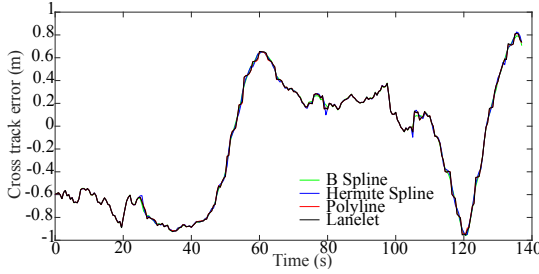


Figure 12: Cross Track errors with a low cost GNSS receiver.

control with the B-spline being the smoothest of the three.

C. Along-track and cross-track errors

For autonomous navigation systems, the localization performances often need to be evaluated in terms of along-track and cross-track errors instead of Euclidean errors. However, the computation of the errors depends on how the curvilinear coordinates are computed. In the previous experimental setup, we also recorded localization data from a low cost Ublox 8T mono-frequency GNSS receiver.

Fig. 11 shows the along-track errors and Fig. 12 the cross-track errors with a low cost GNSS receiver wrt the ground truth computed with the different road models. One can see, as expected, that discontinuity issues for the along-track errors appear for the polyline model. For the cross-track errors, there is no discontinuity, regardless of the road model.

Tab. II shows the mean squared errors of the different models. One can see that these mean squared errors are similar as Fig. 11 and Fig. 12 show.

Because the Ublox GNSS receiver did not provide reliable orientation information, the errors on the curvilinear orientation were not studied.

V. CONCLUSION

This work has presented different models to use discrete maps for localization. These models can be used to compute curvilinear poses, composed by the curvilinear abscissa, the lateral distance and the relative orientation w.r.t. the road.

Low cost GNSS receiver	Polyline	Lanelet	Hermite	B-spline
Along-track errors (m)	0.5745	0.5704	0.5733	0.5692
Cross-track errors (m)	0.5512	0.5499	0.5501	0.5512

Table II: Along-track and cross-track mean squared errors.

Curvilinear poses play an important role for autonomous vehicle navigation.

Different road models have been tested and compared in this work using both real and simulated data. A polyline composed by line segments is a simple to use method for computing curvilinear coordinates. However, it presents many discontinuities and differentiation issues when going from one segment to the next. Nevertheless, it is the only method among the four studied approaches for which all the transformations remain linear. This can be very important in case one needs to derive a motion model, such as a constant velocity model, within the road frame. The Hermite spline and the B-spline models provide smooth continuous and differentiable curvilinear coordinates but are both very computationally costly and are unfortunately not suited for a real-time application. The lanelet road model provides an interesting trade-off between the polyline and the spline models. The curvilinear coordinates are continuous but not differentiable. The non-linearity involved in the lanelet model is not as severe as for the spline-based methods and the computation of the curvilinear coordinates can be done in real-time. We therefore recommend its use in practice.

Acknowledgment: This work was carried out in the framework of Equipex ROBOTEX (ANR-10- EQPX-44-01) and Labex MS2T (ANR-11-IDEX-0004-02). It was also carried out within SIVALab, a shared laboratory between Renault, CNRS and UTC.

REFERENCES

- [1] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient map representation 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] Carl de Boor. *A Practical Guide to Splines*. Springer New York, December 2001.
- [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] E. Héry, Ph. Xu, and Ph. Bonnifait. Along-track localization for cooperative autonomous vehicles. *IEEE Intelligent Vehicles Symposium*, pages 511–516, June 2017.
- [6] C. Katrakazas, M. Qudus, W. H. Chen, and L. Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, November 2015.
- [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] Cristian Constantin Lalescu. Two hierarchies of spline interpolations. Practical algorithms for multivariate higher order splines. *arXiv:0905.3564 [cs]*, May 2009.
- [9] Les Piegel and Wayne Tiller. *The NURBS Book*. Springer Science & Business Media, December 2012.
- [10] 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*, 34:1010–1033, 2017.
- [11] 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.