

# RELIABLE POSITIONING DOMAIN COMPUTATION FOR URBAN NAVIGATION

Vincent Drevelle and Philippe Bonnifait

**Abstract**—Reliable positioning is a key issue for intelligent vehicle navigation. Interval-based positioning methods have shown to be capable of computing relevant confidence domains used for integrity monitoring in environments which are challenging for Global Positioning System (GPS). The approach presented in this paper consists in tightly coupling a GPS receiver with a 3D-map of the drivable area. Interval analysis is employed to solve the constraint positioning problem using contractions and bisections. Integrity is provided through the use of a robust set-inversion scheme applied to a redundant measurement set. If the prior distribution of the measurement noise is known, it is possible to compute confidence domains that correspond to a given integrity risk, which is often set very low out of safety considerations. In this paper we examine a way of validating the proposed approach, using a real experimental dataset and a ground truth equipment. Different tunings of the method, corresponding to different risks, are assessed in terms of availability and integrity in order to compute statistical metrics. Results indicate that this methodology is relevant since the specified risk corresponds to experimental observations.

## I. INTRODUCTION

GPS positioning is a core technology for intelligent vehicle navigation. It provides worldwide positioning with a cost that can be very affordable. In complex environments, like city centers, satellite signals can be blocked (which degrades the geometrical configuration) and can be affected by multipath. Since positioning quality can vary widely, it is important to be able to quantify in real-time the quality of a computed position. One way of addressing this problem is by computing *confidence domains* that quantify position uncertainty. If the confidence domain becomes too wide, the positioning system can simply fire an alert to inform the navigation system that the positioning quality is too bad for the current task. This is the Receiver Autonomous Integrity Monitoring (RAIM) approach [1].

Confidence domains have to take into account measurement errors, which can be classified into two categories: *noise errors* and *faults* i.e. errors that do not correspond to the observation model. Probabilistic frameworks combining estimation (snapshot or sequential, using Kalman filtering for instance) with Detection, Identification and Adaptation (DIA) (see, for example, [2]) are often used. Approaches of this kind yield a point position estimate and then quantify the error distribution around this point.

Another way of addressing this problem is to use set-membership methods which can propagate measurement intervals up to the computation of the solution set. The method that we use in this paper formulates the problem

as a *Constraint Satisfaction Problem* (CSP) combined with iterative set inversion. Using interval analysis and contractors, the positioning confidence domain is computed as the set of positions compatible with all the constraints. This kind of solver adds no risk to the computation. It can provide a disconnected solution set where there is positioning ambiguity (as a result, for example, of non-linearities) and can work in constrained conditions (e.g. with only two satellites in view). Finally, the computation mechanism can be robustified to handle outliers using *q-relaxation* [3]. These methods make it possible to characterize in real-time confidence domains corresponding to a given risk.

The main concern addressed in this paper is verifying that the confidence domains computed using this kind of approach are reliable and relevant (*i.e.* not too pessimistic and the confidence domains computed at the final decision stage correspond to the specified risk). This is the main contribution of this paper. To this end we present a set-membership evaluation methodology for comparing the result from the computed solution set with a ground truth equipment whose level of inaccuracy is also taken into account. Since integrity is closely linked to availability, we report the results of an experimental analysis focusing on these two performance indicators for several given risks.

The paper is organized as follows. We first describe the positioning solver that combines, in a tightly-coupled fashion, GPS pseudorange measurements with a precise 3D road surface model. This solver computes confidence domains in real-time with a constant specified integrity risk on the resulting positioning, by setting appropriate bounds on the GPS measurements. A method for implementing this strategy is presented. An evaluation methodology using a ground truth system and the experimental set-up is then presented in Section 3. Finally, real experimental results for different risk tunings are analyzed in Section 4.

## II. POSITIONING ALGORITHM

The positioning solver used in this paper involves interval analysis and multidimensional intervals or boxes [4]. This kind of set-membership approach is known to be very efficient in terms of complexity and adequacy for real time processing compared to others, like ellipsoidal state estimation [5].

### A. Basic concepts

A box is denoted  $[x] = [\underline{x}, \bar{x}]$ , where vectors  $\underline{x}$  and  $\bar{x}$  are respectively the lower and upper bounds of  $[x]$ . The solver involves constraint propagation and set inversion with subpavings.

When the components of a vector  $x$  are linked by relations or constraints, a constraint satisfaction problem can be defined. This consists in finding the set of solutions  $X = \{x \in [x] | f(x) = 0\}$ , where  $[x]$  is the domain of the variables and  $f(x) = 0$  represents the constraints. Inequalities can also be represented by introducing slack variables [4].

A *contractor*  $\mathcal{C}$  for a CSP is an operator that computes a smaller domain  $[x_c] = \mathcal{C}([x])$  without affecting the solution set, i.e.  $X \subset [x_c] \subset [x]$ . There are many ways to implement a contractor, one of them being the forward-backward contractor based on constraint propagation [6].

A set inversion problem consists in determining the set  $X$  such that  $f(X) \subset [y]$ , where  $[y]$  is a known interval vector of  $m$  measurements. To approximate compact sets in a guaranteed way, *subpavings* can be used. A subpaving of a box  $[x]$  is the union of non-empty and non-overlapping subboxes of  $[x]$ .

Using interval analysis, the solution  $X = f^{-1}([y])$  can be approximated between two subpavings  $\underline{X}$  and  $\overline{X}$  such that  $\underline{X} \subseteq X \subseteq \overline{X}$ . The *SIVIA* (Set Inversion Via Interval Analysis) algorithm can perform this kind of set inversion, by recursively bisecting an initial box [4].

Since we are seeking to characterize the positioning confidence domain, we only need to compute the outer subpaving  $\overline{X}$  of the set that fulfills positioning constraints. Algorithm 1 implements such a SIVIA strategy:  $\overline{X}$  is an outer approximation of the solution set, given an initial domain  $[x_0]$ . This algorithm uses a list of boxes  $\mathfrak{L}$  managed by the *push* and *pull* functions. If  $\mathfrak{L}$  is a queue, the algorithm employs a breadth-first strategy.  $\epsilon$  controls the sharpness of the subpaving  $\overline{X}$ . Boxes larger than  $\epsilon$  after contraction are bisected along the component interval which has the largest dimension.

---

**Algorithm 1** SIVIA(in:  $[x_0], \epsilon$ )

---

```

1:  $\overline{X} := \emptyset$  // empty subpaving
2: push( $[x_0], \mathfrak{L}$ )
3: while  $\mathfrak{L}$  is not empty do
4:    $[x] := \text{pull}(\mathfrak{L})$ 
5:    $[x] := \mathcal{C}([x])$  // contract the box
6:   if width( $[x]$ )  $< \epsilon$  then
7:      $\overline{X} := \overline{X} \cup [x]$ 
8:   else if  $[x] \neq \emptyset$  then
9:      $([x_1], [x_2]) := \text{bisect}([x])$ 
10:    push( $[x_1], \mathfrak{L}$ ); push( $[x_2], \mathfrak{L}$ )
11:  end if
12: end while
13: return  $\overline{X}$ 

```

---

### B. GPS contractor

GPS positioning with *pseudorange* measurements is a *Time of Arrival* method [7]. Pseudoranges are the distances to each visible satellite, offset by an unknown amount due to the offset between the receiver clock and the GPS time. GPS positioning using pseudoranges is thus a four-dimensional problem: along

with the Cartesian coordinates  $(e, n, u)$  of the user in a local frame tangential to the Earth, the user's clock offset  $dt^u$  has to be estimated. Let  $x = (e, n, u, d)$  with  $d = c \cdot dt^u$ .

Measured pseudoranges need to be corrected, to take into account relativity effects and ionospheric/tropospheric delay. Even after the propagation delay corrections provided in the broadcast GPS data have been applied so as to obtain corrected pseudoranges  $\rho_i$ , model errors remains. Moreover, receiver measurement errors also need to be taken into account. Corrected pseudorange measurements are thus represented as intervals  $[\rho_i]$  whose bounds are determined given an integrity risk [8].

Satellite positions  $(e_i^s, n_i^s, u_i^s)$  are computed with broadcast ephemeris data. They are represented as boxes  $[x^s] = ([e_i^s], [n_i^s], [u_i^s])$  to take ephemeris inaccuracy into account.

The location zone computation involves characterizing the set  $X$  of all locations compatible with the measurements and the satellite position intervals. Each pseudorange introduces a constraint on the solution. The constraint introduced by the  $i^{\text{th}}$  corrected pseudorange measurement is represented by the following function:

$$[\rho_i] = \sqrt{([e] - [e_i^s])^2 + ([n] - [n_i^s])^2 + ([u] - [u_i^s])^2} + [d] \quad (1)$$

The contractor for a single pseudorange constraint will hereafter be referred to as *Pseudorange\_contract*( $[x], [\rho], [x^s]$ ). It is a forward-backward contractor, which performs constraint propagation in an optimal order, following the elementary constraint decomposition of the pseudorange measurement function [6].

The constraint for GPS positioning using pseudoranges corresponds to fulfilling the pseudorange constraints for all visible satellites. With  $m$  satellites in use, a GPS contractor is given in Alg. 2.

---

**Algorithm 2**

---

GPS\_contract(in:  $[x], [\rho]_1, \dots, [\rho]_m, [x^s]_1, \dots, [x^s]_m$ )

---

```

1: for  $i := 1$  to  $m$  do
2:    $[x] := [x] \cap \text{Pseudorange\_contract}([x], [\rho_i], [x^s]_i)$ 
3: end for
4: return  $[x]$ 

```

---

### C. Map constraints

3D roads can be used for contracting not only the East and North estimates by making use of the road boundaries, but also altitude. The *drivable space* denotes the area within which the vehicle can travel. For a car, the drivable space can be defined as the surface of the roadway. Obstacles like posts or lane separators are excluded from the drivable space.

The drivable space is represented in 3D by a surface made up of connected triangular facets (Fig. 1). Vertices are represented by their 3D coordinates, while facets are defined by a list of three vertices. We assume that the vehicle moves

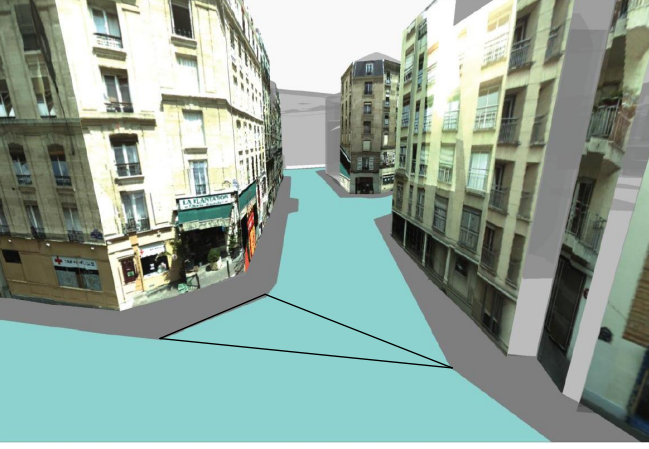


Figure 1: 3D view of the drivable space extracted from a database. A particular triangular facet is highlighted.

only within the represented drivable space, which provides a very strong constraint on the position.

Being located on a 3D triangular face can be expressed by four simple constraints: one constraint representing the facet plane, and three constraints for the edges. The first constraint is simply given by the plane equation obtained from the three vertices. Edge constraints can be expressed by enforcing the positivity of the dot products of vertex-to-point vectors with the corresponding edge normal vectors.

Algorithm 3 presents this contractor (interval vectors  $[a], [b], [c]$  are the coordinates of the facet vertices and the points are in counter clockwise order). The intersection  $\cap$  of two boxes is a box. Since the union of two boxes is not necessarily a box (e.g.  $[1, 2] \cup [3, 4]$ ), let us define the box union  $\sqcup$  which returns the hull of the union of two boxes (e.g.  $[1, 2] \sqcup [3, 4] = [1, 4]$ ).

---

**Algorithm 3** Facet\_contract(in:  $[x], [a], [b], [c]$ )

---

```

1:  $[bbox] := [a] \sqcup [b] \sqcup [c]$ 
2:  $[x] := [x] \cap [bbox]$ 
3: if  $[x] = \emptyset$  then
4:   return  $[x]$ 
5: end if
6: if  $([x_{1:2}] - [a_{1:2}]) \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} ([b_{1:2}] - [a_{1:2}]) \cap [0, +\infty] = \emptyset$ 
   or  $([x_{1:2}] - [b_{1:2}]) \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} ([c_{1:2}] - [b_{1:2}]) \cap [0, +\infty] = \emptyset$ 
   or  $([x_{1:2}] - [c_{1:2}]) \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} ([a_{1:2}] - [c_{1:2}]) \cap [0, +\infty] = \emptyset$  then
7:   return  $(\emptyset)$ 
8: end if
9: return  $[x]$ 

```

---

The whole map constraint is simply the union of the constraints from each facet of the map  $\mathcal{M}$ . Algorithm 4 contracts a box with the entire map. Map-Matching (MM) is often done with polylines at macro-scale level [9] or meso-scale [10]. It consists in selecting the segment of the polylines which is the most likely. Here, the process is rather different

since no explicit selection is needed. Nevertheless and in order to speed up the process, a selection of all the facets (and not only the most likely one) that intersect the prior box is done. This is achieved by the *Extract\_facets* function which returns the set of facets whose bounding box intersects the prior box. In other words, it enables us to focus only on the interesting part of the map.

---

**Algorithm 4** Map\_contract(in:  $[x], \mathcal{M}$ )

---

```

1:  $[xc] := \emptyset$ 
2:  $\mathcal{F} := \text{Extract\_facets}(\mathcal{M}, [x])$ 
3: for each  $f$  in  $\mathcal{F}$  do
4:    $[xc] := [xc] \sqcup \text{Facet\_contract}([x], f.[a], f.[b], f.[c])$ 
5: end for
6: return  $[xc]$ 

```

---

When a previous position is known, road topology is also used to reduce the initial facet candidates as presented in [11].

#### D. Positioning solver

Positioning is performed by using the map contractor described in Section II-C together with the GPS contractor described in Section II-B. We use the SIVIA algorithm (Alg. 1), with the contractor  $\mathcal{C}_{road\_gps}$  implemented in Alg. 5, which successively applies, at each step of its set inversion scheme, the road and GPS contractors until no more contraction occurs.

---

**Algorithm 5**  $\mathcal{C}_{road\_gps}$ (in:  $[x], \mathcal{M}, [\rho]_{1\dots m}, [x^s]_{1\dots m}$ )

---

```

1: repeat
2:    $[x] := \text{Map\_contract}([x], \mathcal{M})$ 
3:    $[x] := \text{GPS\_contract}([x], [\rho]_{1\dots m}, [x^s]_{1\dots m})$ 
4: until no more significant contraction can be done on  $[x]$ 
5: return  $[x]$ 

```

---

#### E. A priori integrity risk computation

The knowledge of the risk associated with the computed confidence domain is of prime importance for reliable positioning. Assuming that the bounds of map error are known and guaranteed, the risk that the computed solution does not contain the true position depends only on pseudorange error. To compute a confidence domain with a given integrity risk  $r$ , the choice of error bounds on pseudoranges is accomplished as follows [12].

Let  $p$  be the confidence in each pseudorange measurement error interval (see Fig. 2):

$$p = \Pr(\rho_i \in [\rho]_i), i \in \{1 \dots m\} \quad (2)$$

Let  $m_{ok}$  be the number of pseudorange intervals that are consistent with ground truth. Assuming pseudorange errors are independent, the probability that all pseudorange intervals are consistent with the truth is:

$$\Pr(m_{ok} = m) = p^m \quad (3)$$

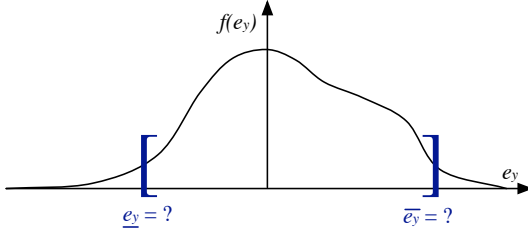


FIGURE 2: Probability density function of the error  $e_y$ . The problem is to find the two values  $\underline{e}_y$  and  $\bar{e}_y$  ( $\underline{e}_y = -\bar{e}_y$ ) corresponding to the confidence value  $p$ .

The positioning solver computes an outer approximation  $\bar{X}(t)$  of the position confidence domain as a subpaving, which is guaranteed to be consistent with the true position  $x(t)$  if all the measurement intervals are consistent with the true error. This way,

$$m_{ok} = m \Rightarrow x(t) \in \bar{X}(t) \quad (4)$$

which leads to

$$\Pr(x(t) \in \bar{X}(t)) \geq \Pr(m_{ok} = m) \quad (5)$$

$$\Pr(x(t) \notin \bar{X}(t)) \leq 1 - \Pr(m_{ok} = m) \quad (6)$$

$$r \leq 1 - p^m \quad (7)$$

For a specified risk  $r$ , (7) can be inverted to find the individual minimum confidence  $p$  on each pseudorange interval. Assuming normally distributed pseudorange errors of variance  $\sigma^2$  (see. Fig 3), one way of setting the error bounds on each pseudorange measurement is then given by

$$[\rho] = [\rho - \alpha\sigma, \rho + \alpha\sigma] \text{ with } \alpha = -\Phi^{-1}\left(\frac{1-p}{2}\right)$$

where  $\Phi$  represents the cumulative distribution function of the standard normal distribution.

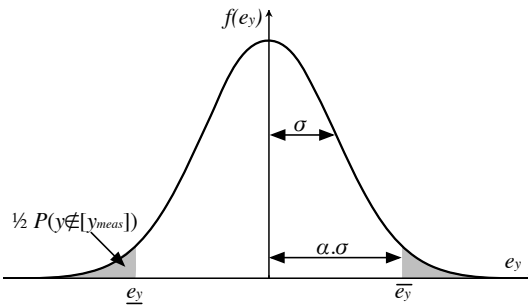


FIGURE 3:  $\pm\alpha\sigma$  bounds for a Gaussian error.

Some values of  $p$  and  $\alpha$  as a function of the number of pseudoranges  $m$  are given in Table I.

### III. EVALUATION METHODOLOGY

The interval-based positioning solver is evaluated based on two criteria: positioning availability and integrity. Other criteria can also be considered depending on the application, as explained in [13].

Table I: Pseudorange intervals risk  $p$  and error bounds for  $r = 10^{-5}$  as a function of the number  $m$  of GPS measurements.

$m$	1	2	3
$p$	$10^{-5}$	$5 \cdot 10^{-6}$	$3.3 \cdot 10^{-6}$
$\alpha$	4.42	4.56	4.65

#### A. Positioning information availability

Navigation requires positioning uncertainty to be small enough to allow position information to be used by the current navigation task. In the same way as a *protection level* (PL) is compared to an alert limit [14], positioning information is declared available if the radius of the computed confidence domain (in the tangent plane) is less than the maximum tolerated position error for the current navigation task. In this study, we have used a 10-meter limit, as suggested in [15].

In this paper, positioning availability will be checked with a box whose width is twice the specified maximum error. Positioning will be declared available when the computed confidence domain fits inside the box.

#### B. Integrity

The positioning solver computes a position confidence domain with an associated integrity risk. This risk corresponds to the case where the true position is not included in the computed confidence domain.

To check positioning integrity, we check that the true position is inside the computed solution subpaving every time position information is declared available.

#### C. Ground-truth uncertainty

Since the radius of the computed location domains can be as small as a few meters, the uncertainty of the ground truth position employed in the integrity evaluation cannot be neglected. Uncertainty about the true position is handled by representing the ground-truth as a box  $[x_{ref}]$  (Fig. 4). The binary integrity test is thus replaced by a 3-state inclusion test:

- “True” means that, despite the ground-truth error, the confidence domain is proven to contain the true position.
- “False” means that the confidence domain is proven not to contain the true position.
- “Indeterminate” means that the ground-truth uncertainty makes it impossible to decide whether the confidence domain contains the true position.

Integrity checking of the solution subpaving is performed using Algorithm 6. If the ground truth box is a subset of a box in the solution subpaving, then integrity is proven to be true. If the subpaving does not contain any box that intersects the ground-truth box, then integrity is false. Otherwise, the result is indeterminate, unless we prove that  $[x_{ref}]$  is a subset of the solution subpaving. To save computation time, we perform a simplified test with the bounding box of the set of boxes that intersect  $[x_{ref}]$ . (This test is not guaranteed, in that it does not prove that the ground-truth box is included in the subpaving.)

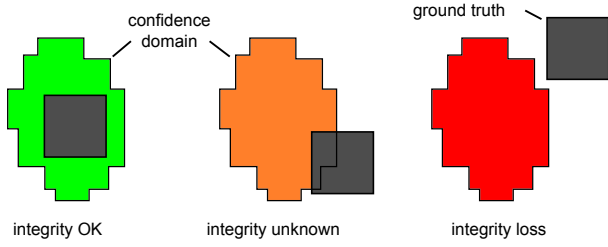


Figure 4: Integrity validation of the computed subpaving with the ground-truth uncertainty displayed in black

---

**Algorithm 6** Integrity\_check( $[x_{ref}], \bar{X}$ )

---

```

1:  $[bbox] := \emptyset$ 
2: for  $[x]$  in  $\bar{X}$  do
3:   if  $[x_{ref}] \subset [x]$  then
4:     return true
5:   else if  $[x_{ref}] \cap [x] \neq \emptyset$  then
6:      $[bbox] := [bbox] \cup [x]$ 
7:   end if
8: end for
9: if  $[bbox] = \emptyset$  then
10:  return false
11: else if  $[x_{ref}] \subset [bbox]$  then
12:  return true
13: else
14:  return indeterminate
15: end if

```

---

## IV. EXPERIMENTAL RESULTS

### A. Experiment

Data were acquired using an experimental vehicle belonging to the *Institut Geographique National* (IGN), as part of a research project known as *CityVIP*. We focus below on a 1 km loop in the 12th arrondissement of Paris, a 5 minute drive around the local town hall (Fig. 5).

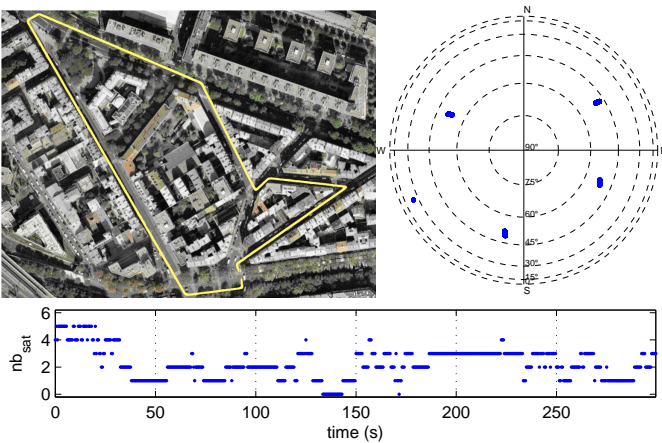


Figure 5: Trajectory around the 12th arrondissement town hall in Paris, skyplot of used GPS satellites, and number of visible satellites with  $\text{SNR} \geq 35\text{dB}$  during the trial.

The 3D map of the drivable space was charted by the *IGN* [16]. It was constructed manually from precise aerial photographs (5 cm horizontal precision, 25 cm in altitude), and converted from its original Lambert93 projection to a local tangent frame, in which GPS positioning is also performed. We use GPS pseudorange measurements acquired at 2 Hz with a *Septentrio PolaRx2* receiver. Satellite positions and pseudorange corrections are computed with the *GPSTk* [17].

To prevent the use of erroneous measurements, all pseudoranges with L1 carrier to noise ratio below 35 dBHz have been filtered out. This efficiently rejects reflected signals since they are attenuated by the use of a polarized antenna. The acquired data are very challenging for GPS positioning. Indeed, after carrier to noise ratio thresholding, there are fewer than 4 satellites in view during 88% of the trial, and fewer even than 3 satellites in view in 56% of the measurement epochs (Fig. 5).

In the positioning risk computation, pseudorange measurement noise is assumed to be Gaussian distributed. One possibility would be to use the User Range Accuracy (URA) broadcast in the GPS data. URAs are figures for each satellite corresponding to upper bounds of the pseudorange noise standard deviation. However, these values are coarsely discretized and all the satellites in the experiment were assigned the lowest representable value, which is  $\sigma_{URA} = 2\text{ m}$ . Therefore, using the URA value would be a too pessimistic estimation of the pseudorange error noise. Many experiments carried out with the same GPS receiver showed that the standard deviation of the pseudorange measurement error (after applying *GPSTk* corrections) is in the order of one meter. This is the value that has been used in the experiments reported in this paper.

Ground truth is provided by a post-processed *Applanix* inertial navigation system. After comparison with two *LandINS* inertial navigation systems also embedded in the vehicle during the data acquisition, the maximum ground truth uncertainty on the horizontal plane was estimated to be less than 1 m [18].

### B. Results

The interval positioning solver was tested over the whole 1 km test loop, for 3 different integrity risk settings ( $r = 10^{-4}$ ,  $r = 0.1$  and  $r = 0.5$ ). We only consider the horizontal component of the estimated location for evaluation, since altitude is constrained by the map. Positioning is declared to be available when the computed confidence domain fits within a 20 m square.

We report results obtained with the same dataset as that used in [19]. The test starts with good satellite visibility, before entering urban canyons with 2 or 3 satellites in view. Accurate positioning can be achieved with only two satellites in view thanks to the road constraint. Sometimes ambiguous positioning occurs at crossroads (i.e. the solver provides several disconnected sets) since two satellites are not enough to select the correct road segment. Note, finally, that the system is able to contract position with only one satellite in view thanks to the receiver clock offset prediction.



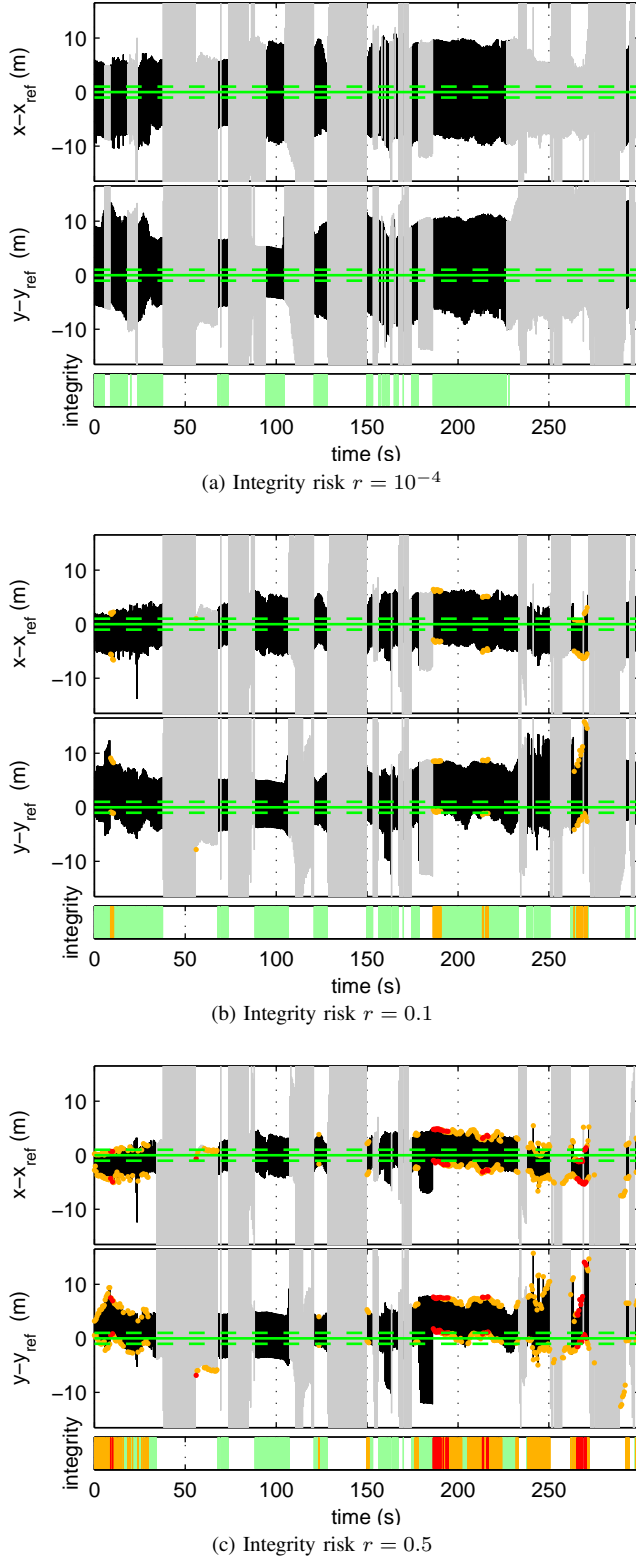


Figure 6: Bounds of the computed position subpaving, availability and integrity. Zero ordinate is ground truth. Grayed-out zones indicate positioning unavailability. Integrity-test results are reported in green=true, orange=indeterminate and red=false.

Fig. 6 shows the bounds of the confidence domains with respect to ground truth for each integrity risk setting. When the confidence domain is too large, positioning is declared unavailable and shown in light gray in the figure. When positioning is available, an integrity indicator is computed, taking ground-truth uncertainty into account. Epochs where the confidence domain is guaranteed to contain the true position are indicated in green. Epochs where the confidence domain is proven not to contain the true position are shown in red. Orange indicates indeterminate cases due to ground-truth uncertainty.

With a very low integrity risk ( $r = 10^{-4}$ ), the confidence domain is consistent with ground truth during the whole trial. The number of epochs in which it cannot be proven that the confidence domain contains the true position increases when computation is done with higher *a priori* risk settings ( $r = 0.1$  and  $r = 0.5$ ). It is interesting to note that the sum of unknown (orange) and loss of integrity (red) is in close agreement with the *a priori* risk setting  $r$ . Table II shows that while the percentage of epochs where integrity is not certainly proven is null with  $r = 10^{-4}$ , this percentage grows to 9% with  $r = 0.1$  and to 52% with  $r = 0.5$ .

Table II: Availability and integrity statistics

integrity risk	$r = 10^{-4}$	$r = 0.1$	$r = 0.5$
availability (20 m square)	37%	54%	56%
integrity ok	100%	91%	48%
unknown	0%	9%	44%
integrity lost	0%	0%	8%

Another interesting fact to note is that setting a very low *a priori* risk is done at the expense of a lower availability. In Table II, positioning availability with  $r = 10^{-4}$  is only 37%, but availability is around 55% with higher *a priori* risks ( $r = 0.1$  and  $r = 0.5$ ).

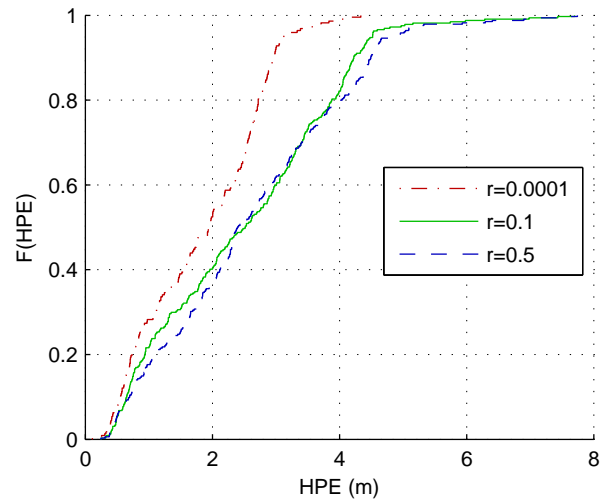


Figure 7: Empirical cumulative distribution functions of the horizontal position error (HPE), for epochs with positioning declared available.

Figure 7 shows the empirical cumulative distribution functions of the horizontal position error (HPE), that have been computed from epochs with positioning declared available. This figure shows better positioning accuracy with  $r = 10^{-4}$  than with  $r = 0.1$  and  $r = 0.5$ . This is mainly due to the difference in positioning availability. With a low integrity risk, position confidence domains are larger. Epochs with bad positioning configurations are thus more likely to be filtered out by the availability threshold with  $r = 10^{-4}$  than with the two higher risk settings. The cumulative distributions of HPE with  $r = 0.1$  and with  $r = 0.5$  are close to each other. This is because nearly the same epochs are declared available with these two risk settings (see Fig. 6). More detailed HPE statistics can be found in Table III, that clearly shows the degradation of positioning accuracy as the integrity risk increases. While the mean error with  $r = 10^{-4}$  is 1.84 m, it goes up to 2.49 m and 2.61 m respectively with  $r = 0.1$  and  $r = 0.5$ .

Table III: Horizontal position error (HPE) statistics, for epochs with positioning declared available

integrity risk	$r = 10^{-4}$	$r = 0.1$	$r = 0.5$
mean HPE	1.84 m	2.49 m	2.61 m
std. dev. HPE	0.96 m	1.44 m	1.47 m
min HPE	0.11 m	0.25 m	0.25 m
max HPE	4.34 m	7.70 m	7.74 m
median HPE	1.94 m	2.52 m	2.40 m
95th percentile HPE	3.14 m	4.52 m	4.92 m

Figure 8 shows the distribution of the true position inside the computed confidence domains. Since the size of the domains changes over time, the results are normalized such that -1 represents the domain's lower bound and +1 the domain's upper bound on the considered axis. Only epochs where positioning was declared available were considered in this figure. With a low risk setting  $r = 10^{-4}$ , ground truth is almost always centered in the confidence domain. Indeed, with such a low risk, the confidence domain is quite large with respect to positioning accuracy. With a higher risk setting like  $r = 0.1$ , the true position is more evenly distributed among the confidence domain. This distribution is not centered in the domain, which is mainly due to the non-isotropy of the map constraint. A longer test with more different road configurations (orientation, crossings, etc) would probably lead to a more centered distribution. Finally, with a very high risk setting  $r = 0.5$ , it can clearly be seen that the true position is frequently outside of the confidence domain.

## V. CONCLUSION

An interval-based positioning algorithm merging GPS pseudoranges with a 3D road surface map in a tightly coupled fashion has been presented in this paper. It computes a position confidence domain with a specified *a priori* integrity risk. Experimental validation was performed with real GPS data and a ground-truth positioning solution to assess the actual integrity risk and positioning availability for three *a priori* risk settings. Integrity checking also takes the ground-truth

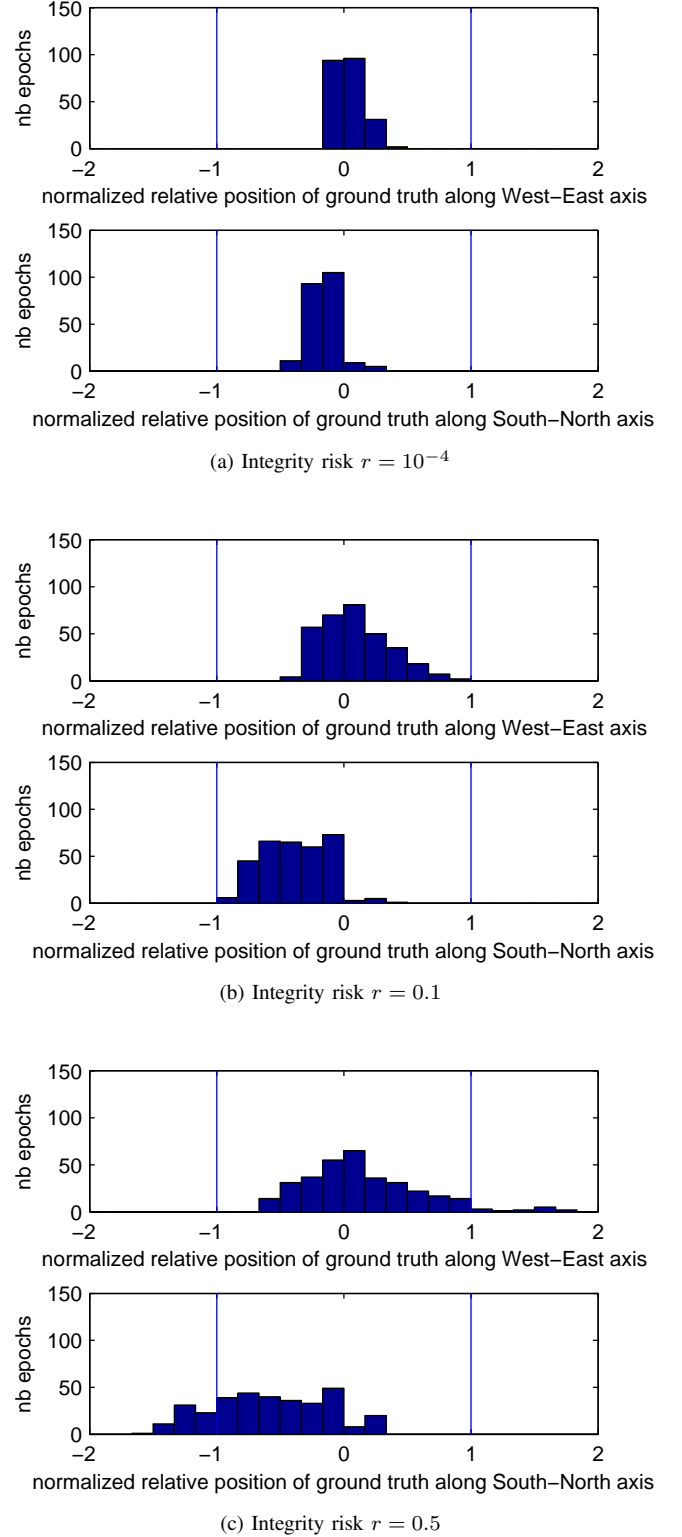
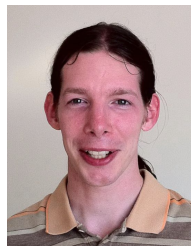


Figure 8: Distribution of the normalized relative position of the true position inside the computed confidence domain, when positioning is declared available. -1 (resp. +1) represents the confidence domain's lower-bound (resp. upper-bound).

uncertainty into account. With a very low risk setting, no integrity losses were observed. With higher risk settings, the proportion of epochs in which integrity has not been validated corresponds closely to the chosen a priori integrity risk: 9% with  $r=0.1$  and 52% with  $r=0.5$ . However, positioning availability, which is conditioned by the size of the confidence domain, decreases as the chosen integrity risk gets smaller. A tradeoff between integrity risk and positioning availability has thus to be found for a given application.

## REFERENCES

- [1] R. G. Brown and G. Y. Chin, "GPS RAIM: calculation of thresholds and protection radius using chi-square methods – a geometric approach," *Global Positioning System*, pp. 155–179, 1997.
- [2] P. J. G. Teunissen, "Quality control in integrated navigation systems," *IEEE Aerosp Electron Syst Mag*, vol. 5, no. 7, pp. 35–41, 1990.
- [3] L. Jaulin, "Robust set-membership state estimation; application to underwater robotics," *Automatica*, vol. 45, no. 1, pp. 202–206, 2009.
- [4] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. Springer-Verlag, 2001.
- [5] B. Polyak, S. Nazin, C. Durieu, and E. Walter, "Ellipsoidal parameter or state estimation under model uncertainty," *Automatica*, vol. 40, no. 7, pp. 1171–1179, 2004.
- [6] F. Benhamou, F. Goulard, L. Granvilliers, and J.-F. Puget, "Revising hull and box consistency," in *Int. Conf. on Logic Programming*. MIT press, 1999, pp. 230–244.
- [7] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications Second Edition*. Artech House Publishers, 2006.
- [8] V. Drevelle and P. Bonnifait, "High integrity GNSS location zone characterization using interval analysis," in *Proceedings of ION GNSS 2009*, 2009, pp. 2178–2187.
- [9] C. Fouque and P. Bonnifait, "Matching raw gps measurements on a navigable map without computing a global position," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 887–898, June 2012.
- [10] R. Toledo-Moreo, D. Betaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 100–112, march 2010.
- [11] V. Drevelle and P. Bonnifait, "Global positioning in urban areas with 3-D maps," in *IEEE Intelligent Vehicles Symposium*, Baden-Baden, 2011, pp. 764–769.
- [12] —, "A set-membership approach for high integrity height-aided satellite positioning," *GPS Solutions*, vol. 15, no. 4, pp. 357–368, 2011.
- [13] I. Skog and P. Handel, "In-car positioning and navigation technologies – a survey," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 1, pp. 4–21, march 2009.
- [14] S. Feng, W. Ochieng, D. Walsh, and R. Ioannides, "A measurement domain receiver autonomous integrity monitoring algorithm," *GPS Solut.*, vol. 10, no. 2, pp. 85–96, 05 2006.
- [15] S. Feng and W. Y. Ochieng, "Integrity of navigation system for road transport," in *World Congress of Intelligent Transportation Systems*, Beijing, Oct. 2007.
- [16] N. Paparoditis, C. Thom, and H. Jibrini, "Surface reconstruction in urban areas from multiple views of aerial digital frame cameras," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, no. Suppl. B3, pp. 43–50, 2000.
- [17] B. Tolman, R. B. Harris, T. Gaussiran, D. Munton, J. Little, R. Mach, S. Nelsen, and B. Renfro, "The GPS Toolkit: Open Source GPS Software," in *Int. Technical Meeting of the Satellite Division of the ION*, Long Beach, California, September 2004.
- [18] D. Betaille, F. Peyret, O. Nouvel, W. Vigneau, G. Duchateau, and H. Secretan, "How to produce a reference trajectory for studying gnss errors in urban environments," in *ENC GNSS*, Toulouse, 2008.
- [19] V. Drevelle and P. Bonnifait, "igps: Global positioning in urban canyons with road surface maps," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 3, pp. 6–18, July 2012.



**Vincent Drevelle** obtained his Ph.D. degree in Computer Science and Engineering at the Université de Technologie de Compiègne (UTC), France in 2011. He graduated as a computer science engineer in 2007; then he obtained a master degree in system and information technologies from the UTC. Since 2008, he has been with Heudiasyc UMR 7253, a joint research unit backed by CNRS and UTC. His current research interests are in set-theoretic methods and interval analysis, for multi-sensor localization of vehicles.



**Philippe Bonnifait** is a professor in the Computer Science and Engineering department of the Université de Technologie de Compiègne (UTC), France. He obtained his Ph.D. degree in automatic control and computer science from the École Centrale de Nantes in 1997. Since 1998, he has been with Heudiasyc UMR 7253, a common research laboratory between UTC and CNRS. His research interests are in Intelligent Vehicles. He is currently working on high integrity positioning and map-matching to retrieve contextual information stored in a navigable

database for Advanced Driving Assistance Systems and autonomous driving.