

Interval-based Fast Fault Detection and Identification applied to Radio-Navigation Multipath

Vincent Drevelle¹ and Philippe Bonnifait²

¹*Université de Rennes 1, IRISA, INRIA Rennes, Lagadic Project, France*

²*Heudiasyc, UMR CNRS 7253, Université de Technologie de Compiègne, France*

SUMMARY

Detection and identification of faulty measurements like multipath is of prime importance for radio-navigation. A method based on interval analysis and constraint propagation is presented in this paper and applied to GPS positioning. This method is able to detect and identify erroneous measurements in real-time simultaneously with the computation of the positioning domain. With bounded-error measurements (GPS pseudorange on visible satellites here), fault-detection is guaranteed not to trigger false alarms. A q -relaxed robust estimation method is studied. By the use of a breadth-first exploration strategy and of measurement consistency counters, the algorithm can promptly signal a detected fault during the set-inversion process. It can also be stopped at any time of the evaluation, and can instantly return the solution subpaving and the list of identified erroneous measurements. We demonstrate that the method is able to handle simultaneous faults as long as they are fewer than the degree of relaxation. The method has been evaluated with real GPS pseudodistances in an urban environment with a low-cost high-sensitivity GPS receiver providing numerous faulty multipath measurements and we report experiments. As the application of this method deals with road vehicles, a 3-D map of the drivable space has also been used to constrain the vehicle location to improve the performance of the method. Copyright © 0000 John Wiley & Sons, Ltd.

KEY WORDS: Fault detection, Set-membership methods, Localization, Intervals

1. INTRODUCTION

It is well known that the estimation of the global position of road vehicles is very sensitive to faulty measurements. In open sky conditions, GPS (Global Positioning System) faults may arise from satellite or control segment failures. Augmentation systems (like WAAS in the USA and EGNOS in Europe) have been designed to protect users against such system failures. Another source of erroneous measurements is the alteration of the signal propagation by the environment close to the user. Indeed, time-of-flight signals are classically assumed to follow a direct path from the satellite to the receiver. However, especially in urban environments where direct path can be frequently blocked by obstacles (such as buildings or trees), the receiver may acquire only a reflected ranging signal.

*Correspondence to: Vincent Drevelle, IRISA Rennes, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France.
E-mail: vincent.drevelle@irisa.fr

It is called a *non line-of-sight* (NLOS) propagation. This measurement is then a fault (also called an outlier), since the observation model is based on the hypothesis of direct propagation. Generally speaking, faulty measurements are sensor data that cannot be explained by the observation model, either because they correspond to extreme noise values, or because they result of phenomena that have not been modeled.

Fault Detection and Identification (FDI) methods are nowadays implemented in some GPS receivers to provide *receiver autonomous integrity monitoring* (RAIM). RAIM is used in aeronautical navigation to detect, identify and eliminate erroneous measurements before computing *Protection Levels* (PL), which are bounds on the positioning error that may result from undetected faults given a target integrity risk. FDI is a key stage in this process and its capability to detect faults with little latency is very important to alert client applications in real-time. FDI is a more and more important stage in positioning integrity since it can be implemented to do adaptation which is a more elaborate processing than a simple fault exclusion. Indeed, FDI can also be used to enable the use of NLOS measurements [1]. When a multipath is detected, the identified corresponding measurement can be tagged before the positioning computation stage. If there are not enough available measurements to do the computation and instead of rejecting the NLOS measurements, the solver can exploit them with adequate observation models that take into account reflections on nearby buildings like, for instance, the “urban trenches model” [2]. Positioning availability is then deeply enhanced.

FDI methods are often based on range residuals [3] or parity space [4]. They rely essentially on exploiting data redundancy. Usually, a first fault detection stage is applied. If a fault is detected, an identification step is performed. The measurement that is an outlier candidate is then removed from the processing and, if enough redundancy remains, the process is iterated until the data are consistent. Finally, a PL confidence domain is estimated under the hypothesis that just one undetected outlier remains after the FDI process. The protection zone is often computed by considering that the minimal detection bias is affecting the measure which has the strongest impact on the position.

Interval-based FDI methods are an interesting alternative to the cited methods since they can perform fault detection and identification *simultaneously* with positioning domain computation. Indeed, interval based methods have successfully been applied to model based diagnosis, as they enable to take model uncertainties into account and can handle nonlinear equations. Interval observers can be used to detect change of operation mode [5], and a guaranteed quasi-linear parameter-varying approximation has been introduced to deal with non-linear systems in [6]. A parity space approach where the parity matrix depends on uncertain parameters has been presented in [7]. Interval-based consistency tests can also be combined with causal reasoning to achieve fault isolation of a complex process [8]. These methods have been employed in various domains, like fault-detection for robot manipulators [9], or height-aided GPS localization [10] with fault-detection results consistent with standard RAIM methods. Other set-membership methods, based on zonotopes for instance, have been used to compute uncertainty in GPS networks [11] and to monitor industrial processes [12].

Moreover, set-membership methods can be robustified in order to handle outliers without providing empty solutions, for linear bounded-error estimation [13], and to perform a distributed estimation in a network of sensors [14].

In this paper, we compute a position domain in which the user is located using an interval based method that is robust to the presence of erroneous GPS measurements. The method performance is increased thanks to the use of a 3D map of the drivable space. Along with the computation of the localization confidence domain, the algorithm enables to detect and identify the faulty measurements. The main contribution of this research is to present and evaluate on real data an algorithm that performs a short response time FDI within a set-inversion process thanks to the use of *measurement compatibility counters*. As soon as a fault is detected, an alert is set with little latency even if the set-inversion has not yet been completed. Moreover, when the computation is stopped because of real-time constraints, the position solution and a list of the identified erroneous measurements are always immediately available. We demonstrate that the method is able to handle simultaneous faults as long as they are fewer than the degree of relaxation. We focus here on a snapshot localization problem (i.e. epoch by epoch) but the method can be applied to measurements that are stored in a limited memory buffer in which time dependency of position is modeled and considered like done in [15] for confidence domains computation. Therefore, in the following of the paper and for the sake of clarity, only an observation model is considered with no evolution model.

The paper is organized as follows. Interval analysis methods for robust bounded-error estimation are first presented. Then, a way to detect and identify outliers from robust set-inversion results is shown, and an algorithm that enables quick interruption of computation with fault detection result is presented. Its properties in terms of detection and isolation are demonstrated. Finally, experimental results are reported with real raw GPS measurements affected by frequent multipath in a challenging urban environment.

2. ESTIMATION USING INTERVAL ANALYSIS

2.1. Interval analysis

Interval analysis [16, 17] involves intervals and their multidimensional extension, *interval vectors* (or *boxes*). In opposition to arbitrary sets, intervals and boxes are easy to represent and manipulate. The set of real intervals $[x] = [\underline{x}, \bar{x}]$ is denoted \mathbb{IR} , and the set of n -dimensional boxes is \mathbb{IR}^n . In this paper, a box is denoted $[x] = [\underline{x}, \bar{x}]$, where vectors \underline{x} and \bar{x} are respectively the lower and upper bounds of $[x]$.

Interval arithmetic allows performing computations on intervals and boxes thanks to the interval extension of classical real arithmetic operators $+$, $-$, \times and \div .

$$\begin{aligned} [x] + [y] &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}], & [x] - [y] &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}], \\ [x] \cdot [y] &= [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})] \end{aligned}$$

In the same way, elementary functions such as *tan*, *sin* and *exp* can be extended to intervals. This is done by returning the smallest interval covering the range of the input through the function.

The image of a box by a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is generally not itself a box, but an arbitrary set. This problem is solved using the so-called *inclusion functions*: The interval function $[f]$ from \mathbb{IR}^n to \mathbb{IR}^m is an *inclusion function* for f if the image of $[x]$ by f is included in the image of $[x]$ by $[f]$,

i.e.

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]).$$

The *box hull* $\square S$ of a set S is the smallest box that includes S . Since the union of boxes is not generally a box, the *box union* operator \sqcup returns the box hull of the union of two boxes: $[\mathbf{x}] \sqcup [\mathbf{y}] = \square([\mathbf{x}] \cup [\mathbf{y}])$.

The width of a box is the width of its longest interval component: $w([\mathbf{x}]) = \max_i w([x_i])$. The volume of a box is defined as the product of the widths of each of its interval components: $\text{volume}([\mathbf{x}]) = \prod_i w([x_i])$.

2.2. Contraction and propagation

When the components of a vector x are linked by relations or constraints, one can define a *constraint satisfaction problem* (CSP). It consists in finding the solution set $X = \{\mathbf{x} \in [\mathbf{x}] | \mathbf{f}(\mathbf{x}) = \mathbf{0}\}$, where $[\mathbf{x}]$ is the domain of the variables and $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ represents the constraints, and can also represent inequalities by introducing slack variables [17].

A *contractor* \mathcal{C} for a CSP is an operator that computes a smaller domain $[\mathbf{x}_c] = \mathcal{C}([\mathbf{x}])$ without affecting the solution set, *i.e.* $X \subset [\mathbf{x}_c] \subset [\mathbf{x}]$. There are many ways to implement a contractor, one of them is the forward-backward contractor based on constraint propagation [18].

2.3. Set inversion and subpavings

The set inversion problem consists in determining the set X such as $\mathbf{f}(X) \subset [\mathbf{y}]$, where $[\mathbf{y}]$ 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 $[\mathbf{x}]$ is the union of non-empty and non-overlapping subboxes of $[\mathbf{x}]$.

Using interval analysis, the solution $X = \mathbf{f}^{-1}([\mathbf{y}])$ can be approximated between two subpavings \underline{X} and \overline{X} such that $\underline{X} \subset X \subset \overline{X}$. The *SIVIA* algorithm allows performing such a set inversion, by recursively bisecting an initial box [17].

Algorithm 1 implements a SIVIA that only computes an outer approximation \overline{X} of the solution set in a given domain $[\mathbf{x}_0]$, since we are seeking to characterize the positioning confidence domain. It uses a list of boxes \mathcal{L} managed by *push* and *pull* functions. If \mathcal{L} is a queue, the algorithm employs a breadth-first strategy. ε controls the sharpness of the subpaving \overline{X} . A contractor $\mathcal{C}([\mathbf{x}])$ is used to apply the constraint from measurements on each box. It may either reduce the size of a box without losing any solution, or return an empty box if the initial box is incompatible with the measurements. Boxes larger than ε after contraction are bisected and enqueued to be processed again.

More extensive implementations of SIVIA that also compute an inner approximation of the solution set can be found in [17].

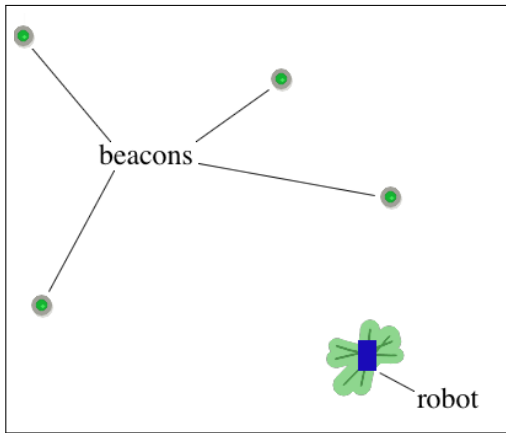
Figure 1 illustrates the use of SIVIA for localization. A robot measures uncertain distances from 4 beacons. In a bounded error context, each measured distance is represented by an interval. Measurements uncertainty is depicted by the thickness of arcs in Fig 1a. Each distance is a constraint on the robot's position. These constraints are implemented in a paver which uses the SIVIA algorithm to compute the set of possible positions (Fig. 1b). The union of red and yellow boxes is an outer approximation of the solution-set. Red boxes are guaranteed to be inside the solution-set.

Algorithm 1 SIVIA(in: $[\mathbf{x}_0], \mathcal{C}, \varepsilon$)

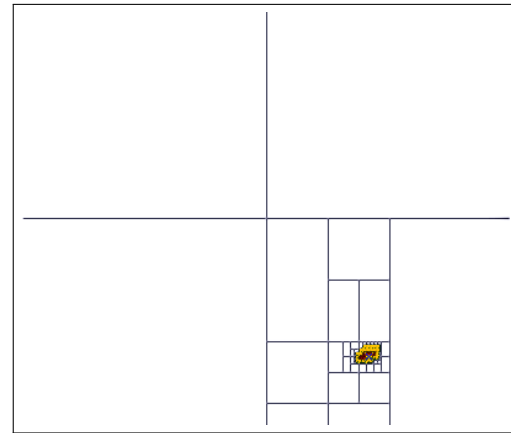
```

 $\bar{X} := \emptyset$  // empty subpaving
push( $[\mathbf{x}_0], \mathcal{L}$ )
while  $\mathcal{L}$  is not empty
   $[\mathbf{x}] := \text{pull}(\mathcal{L})$ 
   $[\mathbf{x}] := \mathcal{C}([\mathbf{x}])$  // contract the box
  if width( $[\mathbf{x}]$ )  $< \varepsilon$  then
     $\bar{X} := \bar{X} \cup [\mathbf{x}]$ 
  else if  $[\mathbf{x}] \neq \emptyset$  then
     $([\mathbf{x}_1], [\mathbf{x}_2]) := \text{bisect}([\mathbf{x}])$ 
    push( $[\mathbf{x}_1], \mathcal{L}$ ); push( $[\mathbf{x}_2], \mathcal{L}$ )
  endif
end
return  $\bar{X}$ 

```



(a) Robot positioning problem with 4 ranging beacons. Width of arcs represents measurement uncertainty.



(b) Position solution-set computed with SIVIA: screenshot of a paving program.

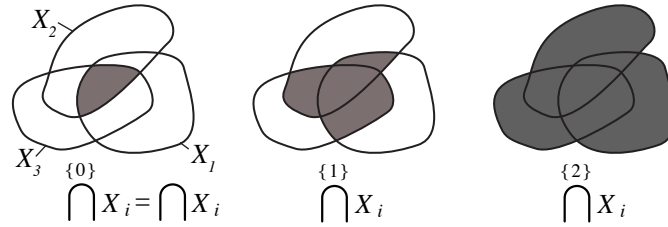
Figure 1. Set-inversion results for a robot localization example.

2.4. Robust set-inversion

The presented set-inversion method is likely to return an empty solution set in the presence of erroneous measurements. Robustness to outliers can be achieved by computing the set of solutions that are compatible with *at least a given number of measurements* instead of the set of solutions compatible with all the measurements.

The q -relaxed intersection of m sets X_i , denoted $\bigcap_{i \in \{1, \dots, m\}}^{\{q\}} X_i$, is the set of points included in at least $m - q$ sets. Figure 2 illustrates this notion, from $q = 0$ that corresponds to the classical intersection up to $q = m - 1$ which is the union. The q -relaxed intersection of m measurement constraints is the set of positioning solutions compatible with at least $m - q$ measurements.

If a contractor \mathcal{C}_i is available for every measurement, then a q -relaxed contractor $\mathcal{C}_{\{q\}}$ can be built. It is presented in Algorithm 2.

Figure 2. q -relaxed intersection of three sets.

Algorithm 2 $\mathcal{C}_{\{q\}}$: q -relaxed contraction (m measures)

for $i = 1$ **to** m { $[\mathbf{x}^i] := \mathcal{C}_i([\mathbf{x}])$ }

 $[\mathbf{x}_c] := \square \left(\begin{array}{c} \{q\} \\ \bigcap_{i \in \{1, \dots, m\}} [\mathbf{x}^i] \end{array} \right)$
return $[\mathbf{x}_c]$

Figure 3 shows the results of the presented SIVIA algorithm with the q -relaxed contractor. Four constraints are provided by ranging beacons, one of which is faulty and gives an erroneous distance measurement. SIVIA is first employed with the $\mathcal{C}_{\{1\}}$ contractor to compute an outer approximation of the 1-relaxed intersection of the three constraints. The obtained solution subpaving is consistent with ground truth. Two extra constraint relaxations are performed to illustrate the method behavior. When using $\mathcal{C}_{\{2\}}$, three disconnected solution subsets appear and, when relaxing a third constraint, the solution set becomes very large. The provided domains always contain the real solution since the number of faults is less than the degree of relaxation.

2.5. GOMNE

Using the q -relaxed intersection for robust set inversion requires to know an upper-bound on the number of erroneous measurements and ideally a rough idea of this number in order to have an accurate estimation as shown by the example above. Since setting an a-priori bound is not always possible, another strategy consists in estimating the number of outliers during the computation. This strategy is known as the *Guaranteed Outlier Minimal Number Estimator* (GOMNE) [19]. Computation starts with no constraint relaxation, and the number q of relaxed GPS measurements is increased until a non-empty solution is found for $q = q_{min}$. Indeed, an empty solution-set with a q -relaxed set-inversion indicates that there are at least $q + 1$ outliers in the measurements vector.

The number of outliers q_{min} detected by GOMNE is only guaranteed as a lower bound, but their actual number can be larger. Thus, in case of undetectable errors, the computed solution may still be wrong after GOMNE iterations. This is the case in Fig. 4b, where GOMNE fails to compute a position solution-set consistent with the ground truth. Indeed, there is an erroneous measurement (4a), but the error is small enough so that there is no inconsistency between the 4 distance constraints. The solution-set with $q = 0$ is thus non-empty, and GOMNE fails to find the presence of an outlier. A common practice is to first determine a lower bound q_{min} of the number of outliers with GOMNE, and then perform a new robust set-inversion with a $(q_{min} + r)$ -relaxed intersection, where r is the number of undetected errors to take into account. Figure 4c shows the

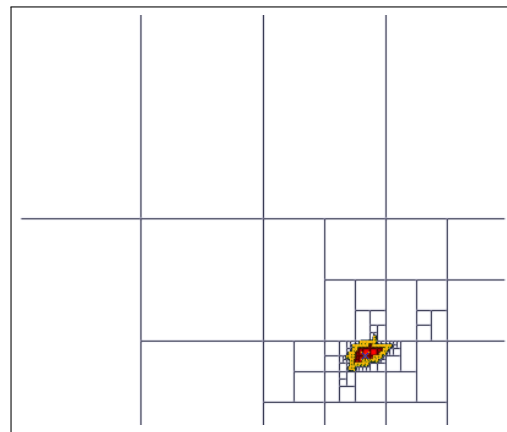
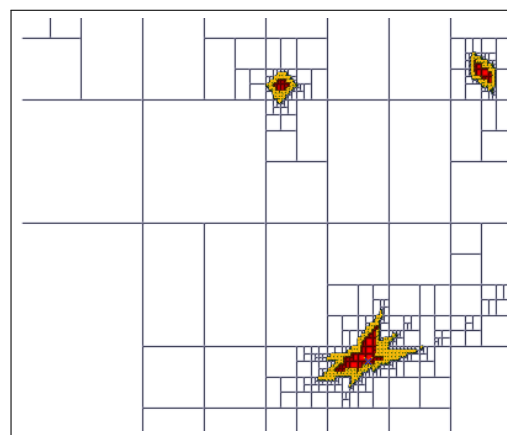
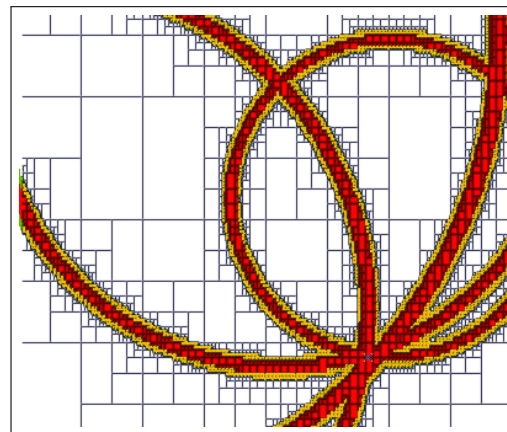
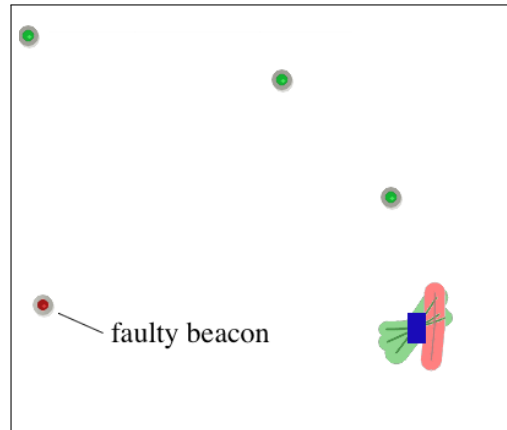
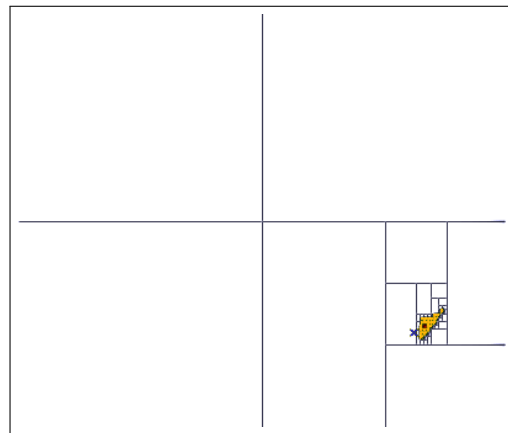
(a) $q = 1$ (b) $q = 2$ (c) $q = 3$

Figure 3. Results of q -relaxed set-inversions in the localization problem with 4 ranging beacons.

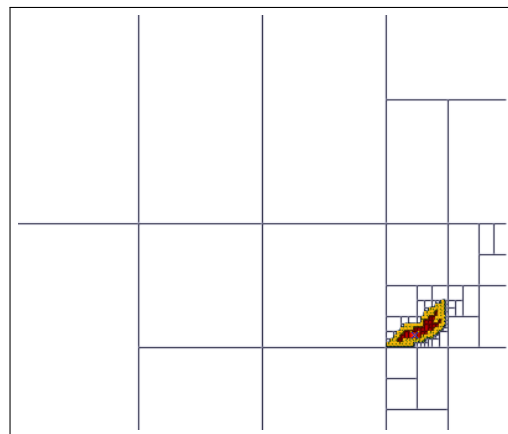
result of GOMNE with a margin of one undetected outlier. The obtained solution subpaving is here consistent with ground truth.



(a) Localization problem with one faulty beacon.



(b) Failure of GOMNE: the faulty measurement has not been detected and the real location (denoted by the cross) is not inside the computed domain.



(c) GOMNE with an additional margin of one undetected outlier. The real location is inside the computed domain.

Figure 4. Results of GOMNE in a localization problem with an erroneous measurement.

Whatever the robust set-membership estimation technique used (q -relaxed intersection with *a priori* known number of outliers or GOMNE with additional undetected errors margin), the q -relaxed mechanism is the core part. For this reason, the following focuses on q -relaxed intersection. Moreover, this technique fits well with real-time constraints.

3. FAULT DETECTION AND IDENTIFICATION

After having reviewed some robust estimation techniques, let us now study how they can be enhanced to perform the FDI process simultaneously with the set-inversion.

3.1. Principle

The basic idea of the method is to compute a solution set and then to analyze the compatibility of the measurements with this set.

In practice, it is necessary to compute a solution with robustness. Otherwise, the solution can be empty, which prevents fault identification. As said before, the robust method considered is the q -relaxed set inversion, which enables computing a solution-set robust to a specified number of outliers. This solution set is represented by an outer subpaving $\bar{X} \supset X$.

The method that we propose to detect faults is to check if the boxes of the solution subpaving are compatible with each measurement. It is also possible to identify the faulty measurements. Geometrical configuration and measurement uncertainty make detection and identification not always possible.

The compatibility of every box of the subpaving is evaluated through inclusion tests with every elementary measurement. A box $[\mathbf{x}]$ is said incompatible with the measurement $[y_i]$ and the observation model f_i if $[f_i]([\mathbf{x}]) \cap [y_i] = \emptyset$. A box $[\mathbf{x}]$ is said incompatible with the measurement vector $[\mathbf{y}]$ if it is incompatible with at least one component of $[\mathbf{y}]$.

Theorem 1. *The presence of an outlier is detected when the outer subpaving only contains boxes incompatible with the measurement vector.*

Proof

In the absence of outliers, every measurement respect the observation model, i.e $\mathbf{f}(x_{true}) \in [\mathbf{y}]$ where x_{true} denotes the real position.

$$\begin{aligned}
 \mathbf{f}(x_{true}) \in [\mathbf{y}] &\Rightarrow x_{true} \in \mathbf{f}^{-1}([\mathbf{y}]) \\
 &\Rightarrow x_{true} \in \bigcap_{i \in \{1 \dots m\}}^{\{q\}} f_i^{-1}([y_i]) \\
 &\Rightarrow x_{true} \in \bar{X} \\
 &\Rightarrow \exists k \mid x_{true} \in [\mathbf{x}]_k, [\mathbf{x}]_k \text{ being a box of the subpaving} \\
 &\Rightarrow \exists k \mid [\mathbf{f}]([\mathbf{x}]_k) \cap [\mathbf{y}] \neq \emptyset
 \end{aligned}$$

By contraposition, if $\forall k \mid [\mathbf{f}]([\mathbf{x}]_k) \cap [\mathbf{y}] = \emptyset$ then an outlier is present. \square

Theorem 2. A measurement is identified as an outlier when the outer subpaving only contains boxes incompatible with this measurement. This holds true as long as the number of outliers n_{out} remains lower or equal to the degree q of the q -relaxation.

Proof

Let us prove it by contraposition, and consider a measurement $[y_i]$.

As $n_{out} \leq q$, $x_{true} \in \bar{X} \Rightarrow \exists k \mid x_{true} \in [\mathbf{x}]_k \Rightarrow \exists k \mid f_i(x_{true}) \in [f_i]([\mathbf{x}]_k)$.

Moreover, since $[y_i]$ is not an outlier, $f_i(x_{true}) \in [y_i]$.

The boxes $[f_i]([\mathbf{x}]_k)$ and $[y_i]$ share a common point, so $\exists k \mid [f_i]([\mathbf{x}]_k) \cap [y_i] \neq \emptyset$.

By contraposition, if $n_{out} \leq q$ and $\forall k \mid [f_i]([\mathbf{x}]_k) \cap [y_i] = \emptyset$, then $[y_i]$ is an outlier. \square

Please notice that no hypothesis on the number of outliers has been done for detection, and that only an hypothesis on the maximum number of outliers is needed for identification. The method is therefore able to manage several simultaneous outliers.

Figures 5, 6 and 7 contain detection and identification tables on pedagogical examples. These tables report the results of compatibility tests for each box, with incompatible boxes represented by zeroes. The last row corresponds to FDI results for the solution subpaving.

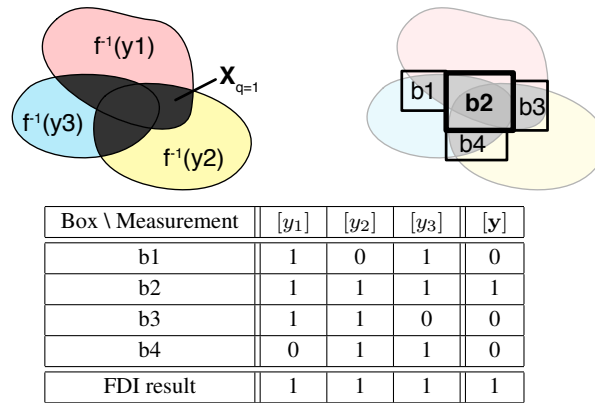


Figure 5. Detection and identification table with no erroneous measurement. The final row describes if the measurements are compatible with the computed subpaving.

In Fig. 5, the compatibility of a box with $[y]$ (last column of the table) is basically computed using a logical AND between the elements of the corresponding row. For instance, box b_2 is compatible with the three measurement intervals and box b_1 is compatible with the single measurements $[y_1]$ and $[y_3]$, but not with $[y_2]$ nor all the measurements together $[y]$. The FDI results for the solution subpaving are obtained by applying a logical OR between the elements of each column, zeroes corresponding to detection in the last column and identification in the others. Since the lower-right element equals one, no erroneous measurement is detected.

In Fig. 6, there is no box in the subpaving that can fulfill the constraints of all the measurements since the lower-right element equals zero. The presence of an outlier is thus detected. However, since there is at least a box compatible with each measurement in the solution subpaving, it is not possible to identify the faulty measurement. The detection and identification table contains also extra information. For instance, if there is no more than one outlier at a time, one can conclude that the faulty measurement is either $[y_2]$ or $[y_3]$.

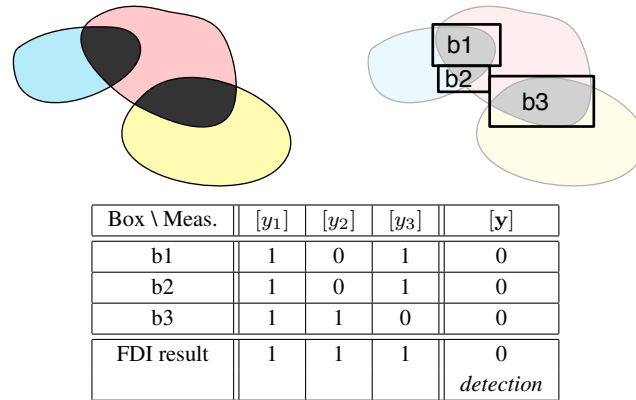
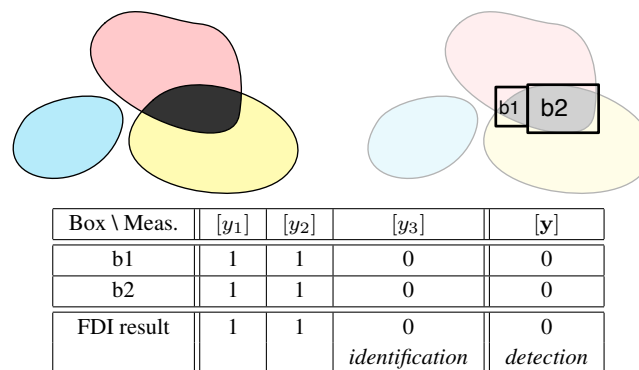


Figure 6. Detection and identification table with an outlier detected.

Figure 7. Outlier identified: $[y_3]$

In Fig. 7, the presence of an erroneous measurement is detected like in Fig. 6. Moreover, the outlier can be identified in this particular case. Indeed, there is no box in the subpaving that can satisfy the constraint from measurement $[y_3]$. The latter is thus identified as the faulty measurement.

3.2. Optimized implementation

One of the main drawbacks of the previous method is that the processing is long to interrupt, since a post-processing has to be performed on the solution subpaving to carry out fault identification. It is unfortunate, as the set inversion process can be interrupted at any time to quickly provide an approximate solution. The ability to provide a solution with a reduced and guaranteed latency is a basic requirement for real-time systems.

We present in this section a way to prepare as much as possible the fault identification process during the exploration of the solution space. The idea is to use *measurement compatibility counters*. As it slows down each iteration of the set inversion algorithm, this implementation reduces the number of bisections per second. However, as the bisection loop terminates, only few quick operations are needed to get the solution subpaving and the list of identified outliers.

RSIVIAExtCount (Alg. 3) computes a subpaving that covers the q -relaxed solution, and, in the meantime, keeps track of the compatibility of the subpaving with the measurements through the use of counters. Set-inversion can be interrupted at any time, and outlier detection and identification can

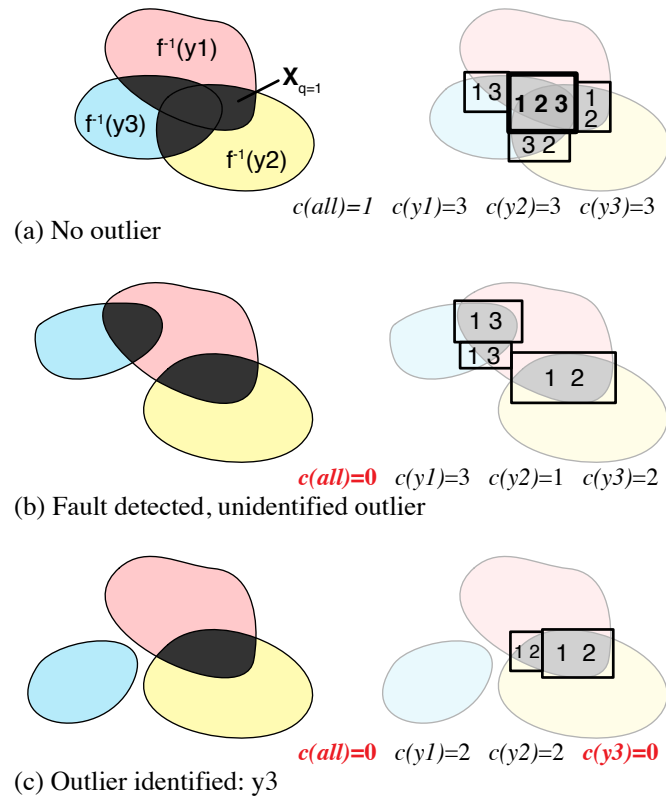


Figure 8. Outlier detection and identification with compatibility counters

then be done by checking the counters values. To reduce the number of inclusion tests, each box is augmented with a compatibility bit-field, which indicates which measurements are compatible with it.

When there is no solution (i.e. an empty set), there are more outliers in the measurements than q , which is the maximum number anticipated. Each counter equals zero in this case. If one wants to get a non empty result, it is possible to restart the estimation process with more relaxation (e.g. $q := q + 1$). This is the GOMNE strategy presented in Section 2.5.

Otherwise, the values of the counters are examined to check the presence of outliers. If the global counter c_{all} is equal to zero, then the presence of an outlier is detected, since there is no box in the solution that is compatible with all the measurements. One can notice that c_{all} corresponds to the lower-right element of a detection and identification table (see Fig. 6). It is then possible to check if an outlier has been identified, simply by studying the values of individual measurement compatibility counters c_i , which keep track of the number of boxes in the subpaving that are compatible with each measurement. If the counter c_i is null, then the measurement y_i is identified as an outlier, like in the last row of a detection and identification table (see Fig. 7). Figure 8 shows the counter values and outlier detection results after a 1-relaxed set inversion on simple examples.

This implementation of fault detection and identification inline with set-inversion not only enables reducing the latency when the computation is stopped, but also monitoring outliers during the computation by instantaneously setting a “fault-detected” flag to the client application (Fig. 9). This is another advantage of this approach since it provides valuable information for online optimization

Algorithm 3 RSIVIAExtCount(in: $[\mathbf{x}], \{C_i, i = 1 \dots m\}, [\mathbf{f}], [\mathbf{y}], q, \varepsilon$; out: $\bar{\mathbb{S}}_x$)

```

for  $i = 1$  to  $m$  { // Initialize measurements compatibility counters
.    $c_i := 1; c_{all} := 1; [\mathbf{x}].c_i := 1$ 
}
 $\mathcal{L} := \{[\mathbf{x}]\}$ 
 $\bar{\mathbb{S}}_x := \emptyset$ 
while  $\mathcal{L} \neq \emptyset$  and not timeout {
.    $[\mathbf{x}] := \text{pull}(\mathcal{L})$  // Pick a box from the list
.   for  $i = 1$  to  $m$  { // q-relaxed contraction with measurements
.      $[\mathbf{x}^i] := C_i([\mathbf{x}])$ 
.   }
.    $[\mathbf{x}_c] := \square \left( \bigcap_{i \in \{1, \dots, m\}}^{q} [\mathbf{x}^i] \right)$ 
.   for  $i = 1$  to  $m$  { // Update inclusion-test results bit-field
.     if  $[f_i]([\mathbf{x}_c]) \cap [y_i] \neq \emptyset$  then  $[\mathbf{x}_c].c_i := 1$ 
.     else  $[\mathbf{x}_c].c_i := 0$ 
.   }
.   // Update counters
.   if  $[\mathbf{x}_c] \neq \emptyset$  then {
.     if  $w([\mathbf{x}_c]) < \varepsilon$  then { // Box is too small
.        $\bar{\mathbb{S}}_x := \bar{\mathbb{S}}_x \cup [\mathbf{x}_c]$ 
.       for  $i = 1$  to  $m$  { // Update compatibility counters
.          $c_i := c_i + [\mathbf{x}_c].c_i - [\mathbf{x}].c_i$ 
.          $c_{all} := c_{all} + \prod_i [\mathbf{x}_c].c_i - \prod_i [\mathbf{x}].c_i$ 
.       }
.     }
.     else { // Bisect and enqueue sub-boxes
.        $([\mathbf{x}_1], [\mathbf{x}_2]) = \text{bisect}([\mathbf{x}_c])$ 
.        $\text{push}([\mathbf{x}_1], \mathcal{L}); \text{push}([\mathbf{x}_2], \mathcal{L})$ 
.       for  $i = 1$  to  $m$  { // Update compatibility counters (2 boxes)
.          $c_i := c_i + 2 \cdot [\mathbf{x}_c].c_i - [\mathbf{x}].c_i$ 
.          $c_{all} := c_{all} + 2 \prod_i [\mathbf{x}_c].c_i - \prod_i [\mathbf{x}].c_i$ 
.       }
.     }
.   }
.   else {
.     for  $i = 1$  to  $m$  {
.        $c_i := c_i - [\mathbf{x}].c_i$ 
.        $c_{all} := c_{all} - \prod_i [\mathbf{x}].c_i$ 
.     }
.   }
. }
. }
// Compute the solution set
 $\bar{\mathbb{S}}_x := \bar{\mathbb{S}}_x \cup \mathcal{L}$ 
if  $c_{all} = 0$  then outlier detected
for  $i = 1$  to  $m$  {
.   if  $c_i = 0$  then  $[y_i]$  is an outlier
}

```

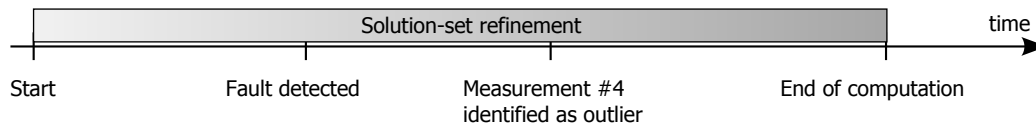


Figure 9. Chronogram of the RSIVIAExtCount algorithm execution in presence of an erroneous measurement.

of the set-inversion computation. Indeed, once a measurement has been identified as an outlier, computing a q -relaxed solution set with all the measurements is equivalent to compute a $(q - 1)$ -relaxed solution with all the measurements but the identified outlier. The robust set inversion algorithm can therefore dynamically discard measurements as they are identified as outliers and thus speed up the computation by reducing the number of constraints to satisfy.

4. APPLICATION TO GPS MULTIPATH FDI

In this section, we report experimental tightly-coupled GPS and map positioning results obtained with a real-time implementation of the algorithm in C++.

4.1. Tightly coupled GPS with map constraints

A tightly coupled approach consists in using raw time-of-flight measurements (pseudoranges affected by the unknown receiver clock offset) from a GPS receiver and performing data fusion for localization. In this paper, a low-cost high-sensitivity GPS receiver (*a uBlox LEA4T*) provides the pseudoranges used to constrain the estimate of the user position. Satellites positions are computed using broadcast ephemeris. Measurement noise is taken into account by representing pseudoranges with intervals. A common practice (not used in this paper) to reduce the number of erroneous GPS measurements consists in filtering out pseudoranges whose signal-to-noise ratio (SNR) is too low, and only keeping pseudoranges from satellites whose elevation is sufficiently high. Without any additional information, the pseudoranges from at least four satellites are required for snapshot positioning. Fault detection is possible with five satellites, and at least six satellites are needed for fault identification.

Tightly coupled fusion enables to reduce the required number of GPS measurements, by using other information in the localization process. Assuming the vehicle only evolves on roads, a strong constraint on position is provided by a 3D map of the drivable space (charted by the french *Institut Géographique National*). The 3D map is a triangle mesh model of the surface of the road.

Positioning is done with the previously presented robust set-inversion algorithm, where a first contractor enforces the road constraint, and a q -relaxed GPS contractor is made from individual GPS pseudorange constraints (Fig. 10). The contractors for GPS and 3D map are respectively based on interval constraint propagation and on a polygon clipping algorithm, and can be found in [15].

4.2. Computation frame

Since intervals are used to characterize the solution set by a subpaving, results are dependent on the coordinate system. An efficient choice is to perform the computations in a “East, North, Up” local

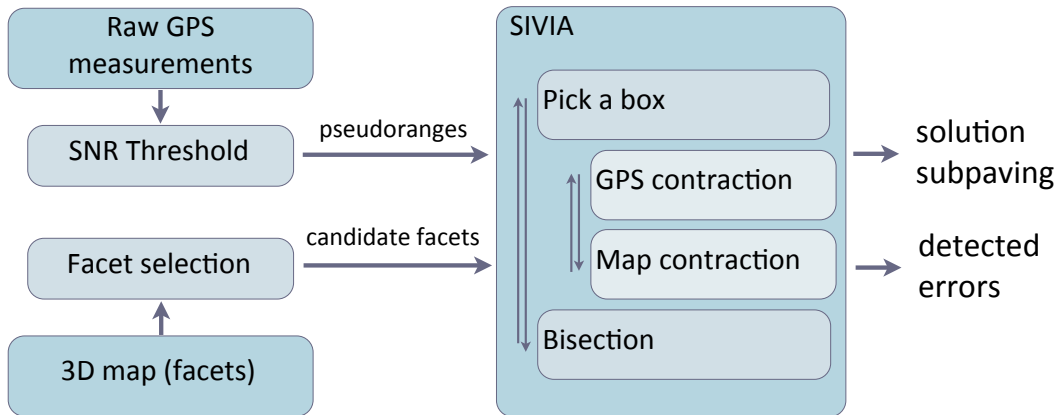


Figure 10. Tightly-coupled positioning with GPS and map constraints

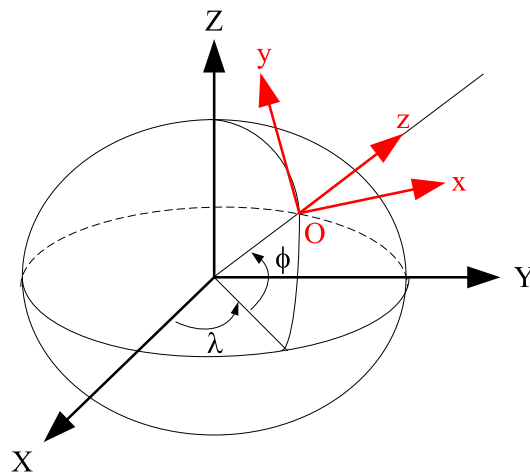


Figure 11. Working frame used to do the computation: x is East, y is North and z is Up.

tangent frame (See Fig. 11) that enables very efficient altitude contraction with the map constraint since roads have often little slope. This frame is kept fixed for a navigation area of few kilometers. An optimization with respect to the interval approach is to align the x -axis with the direction of the current road. It has the advantage to align one side of every facet with the frame but this strategy is time consuming, because it needs to change continuously the coordinates of the vertices of the map and those of the satellites. We believe it is preferable to save processing power to refine set-inversion rather than doing such a coordinates transformation.

4.3. Experiments

The dataset consists in three loops around the *mairie* of the 12th arrondissement in Paris, covering 3 kilometers in 16 minutes (Fig. 12). It is very challenging for GPS since the narrow streets with high buildings prevent good reception of the satellites' signals, and are source of erroneous measurements due to multipath and non-line-of-sight signal propagation. In order to test the presented algorithm in the presence of a lot of outliers, we set the pseudorange SNR filtering threshold to a very low value of 20 dBHz (the standard being at 37 dBHz). Pseudorange measurements errors were modeled by

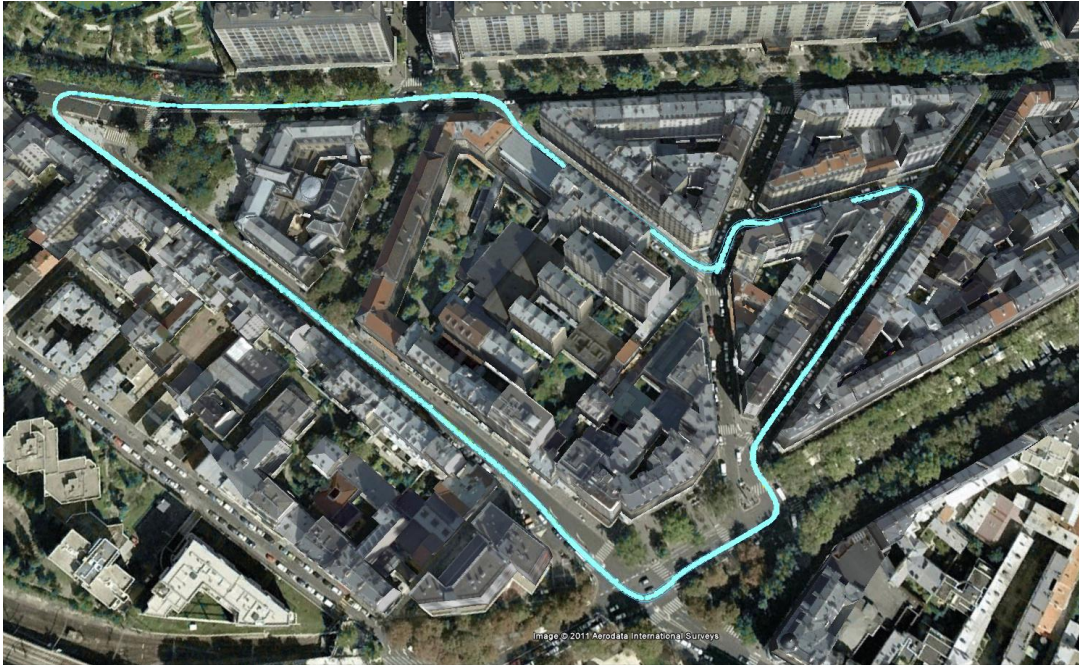


Figure 12. Path followed around the *mairie* of the 12th arrondissement in Paris.

± 3 m intervals which are relevant for the sensor used as shown in [20]. Please notice that there is no hypothesis on the zero-mean or the whiteness of the noise as often done when using Kalman filtering.

In order to evaluate the accuracy of the computed estimates, an *Applanix* inertial navigation system has been used as ground truth.

4.4. Experimental results

The algorithm is able to report a lower bound of the number of outliers by using the GOMNE strategy. This is the number of detected outliers shown in Fig. 13 (top of the bars) and in the left histogram of Fig. 14. At some difficult parts of the trial, *e.g.* at $t=510$ s, the algorithm detects the presence of at least 3 simultaneous outliers in the GPS measurements.

Outliers identification is performed with the compatibility counters presented in the previous section. Figure 15 reports the values of each measurement compatibility counter for 8 epochs between $t=385$ s and $t=420$ s. For the sake of clarity, the counter values are normalized by the number of boxes in the solution subpaving. Identification of an erroneous measurement occurs when the corresponding counter value is zero. The figure shows identification of one outlier at $t=385$ s and $t=410$ s, and of two simultaneous outliers are at $t=400$ s. The number of identified outliers at each epoch is reported by the filled bars of Fig. 13, and summed up in the right histogram of Fig. 14.

Figure 16, illustrates the successful detection and identification of two simultaneous outliers at time $t=400$ s. The lines-of-sight of the satellites corresponding to the identified wrong measurements are indicated by thick red lines.

Due to the lack of redundancy and due to satellite geometry, fault detection may fail, and even lead to wrong exclusion. This happens at time $t=445$ s where a single fault is detected and identified

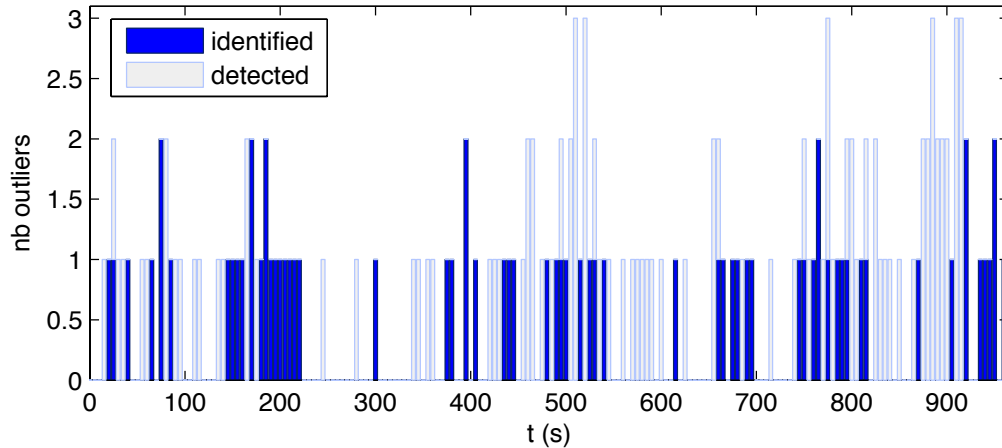


Figure 13. Number of detected (top of the bars) and identified (filled bars) outliers at each epoch.

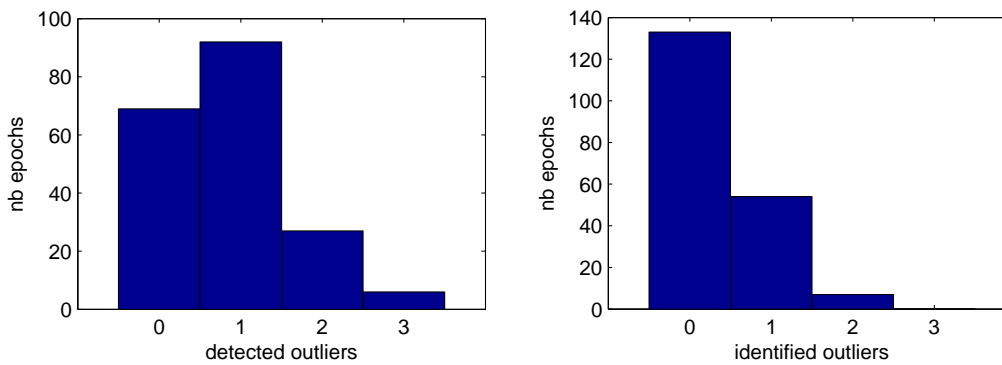


Figure 14. Histograms of the number of detected (left) and identified (right) outliers during the trial.

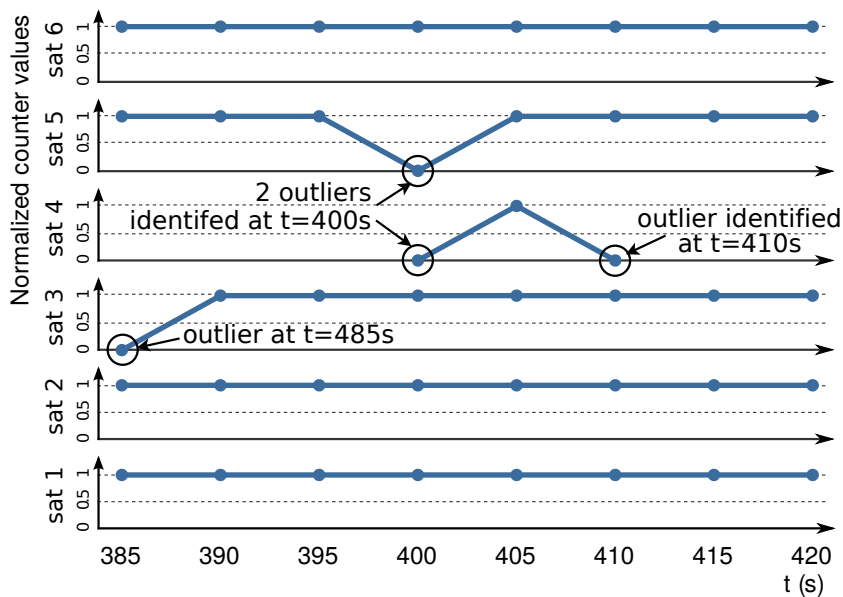


Figure 15. Measurement compatibility counters values (normalized by the number of boxes) of each satellite measurement for 8 epochs between $t=385$ s and $t=420$ s. Erroneous measurements are identified at $t=385$ s and $t=410$ s, and two simultaneous outliers are identified at $t=400$ s.

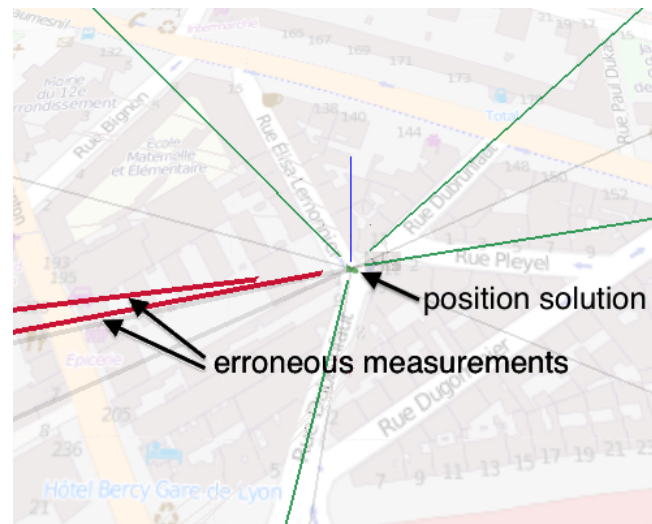


Figure 16. 3D view of the position solution at $t=400$ s. The solution subpaving is in green. Lines-of-sight to the used satellites are represented. Two outliers have been identified, the corresponding lines-of-sight are shown in thick red lines.

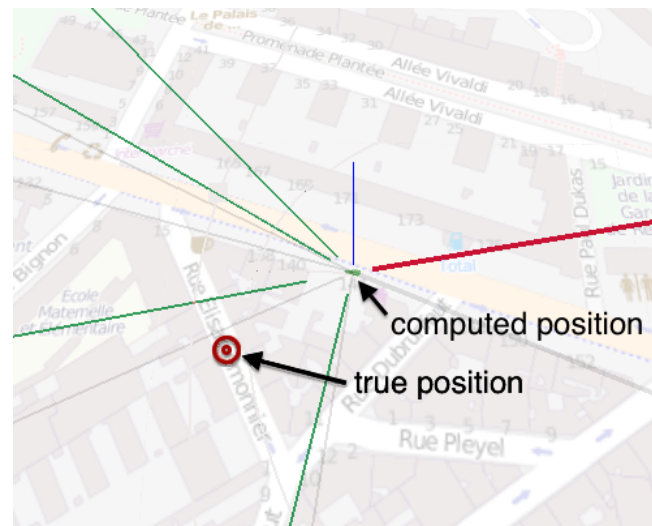


Figure 17. 3D view of the position solution at $t=445$ s, where since 4 out of 5 measurements are erroneous. Fault identification fails and reports only a single erroneous measurement (in thick red). The circle represents ground truth position.

with the measurements compatibility counters, but leads to an erroneous solution set (Fig. 17). In fact, at that particular epoch, four out of the five received signals are very weak, which indicates that the corresponding pseudoranges are erroneous. In a standard setting, these measurements would have been filtered out by the SNR threshold. With 4 wrong measurements out of 5, it is not possible to compute a solution consistent with the ground truth.

It is very difficult to obtain a ground truth for multipath detection. We tried to implement the method presented in [20] but, in the strong canyon-like environment of the reported trial, the method failed often because there were not enough satellites in view and too many simultaneous multipath. Therefore, we study hereafter the accuracy of the computed solution. Indeed, a good accuracy is closely linked to a good FDI stage. As the algorithm computes a solution in the form of a set, a

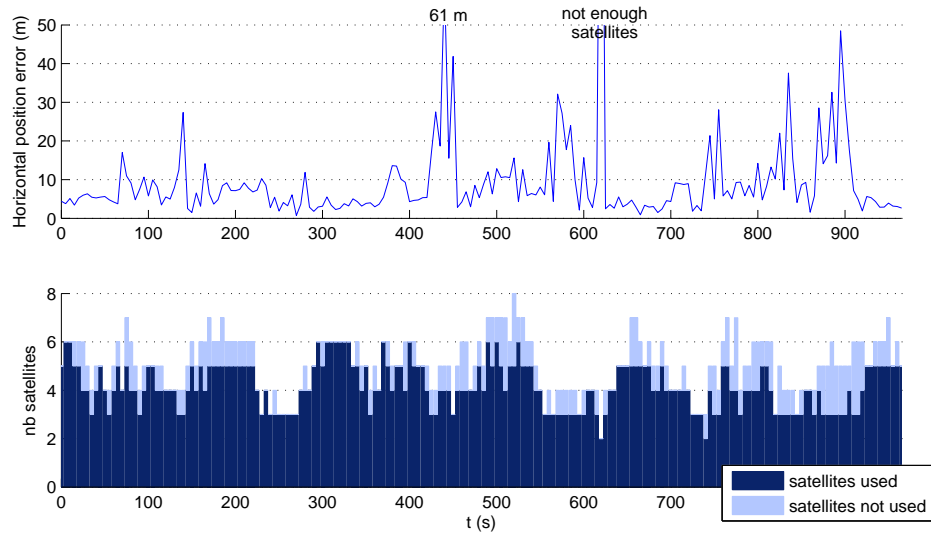


Figure 18. Northing and easting error (in meters) when using the subpaving's center of gravity as a position estimate. The lower plot shows the total number of GPS measurements available at each epoch, detailing the number of used (dark-blue part of the bars) and not used GPS constraints (number of detected faults, light-blue).

point estimate of position is then useful. The center of gravity of the solution subpaving is often used for point estimate. This computation can be performed in real-time with the same philosophy as the FDI method (please see the appendix).

Fig 18 represents positioning error in the north and east coordinates, with respect to the position solution of the *Applanix* inertial navigation system. The mean horizontal positioning error during the whole trial is 9.47 m, with a median of 5.78 m. Large errors in Fig 18 are mainly due to the lack of measurements (*e.g.* at $t=630$ s with only two measured pseudoranges) which leads to disconnected solution sets representing multiple localization hypotheses. Another source of error is the presence of too many outliers at particular epochs since the SNR threshold for measurement rejection was set very low (*e.g.* at $t=445$ s).

Based on the results of Fig 18, one can see that the errors remain acceptable although few satellites are in sight (in normal operation, 4 satellites are necessary to compute a position fix). It can also be noticed that the relaxed (not used) satellites are frequent which corresponds well a typical behavior of a high sensitivity low-cost GPS receiver. We therefore believe that the proposed FDI method behaves quite well.

5. CONCLUSION

An interval-based position estimation method with simultaneous fault detection and identification has been presented. It also enables computing a localization domain from a few GPS measurements with several simultaneous outliers thanks to the help of a 3D map. An optimized implementation with measurement compatibility counters has been presented and tested on a sequence of real GPS data in a very difficult urban environment. The method is able to compute a position and to successfully detect and identify outliers when GPS measurement redundancy is high enough.

However, it may fail to detect a fault when there are not enough GPS measurements (at least 3 satellites are required, then detection performance depends if the error moves the computed position out of the drivable space or not), or identify a good measurement as a fault when there are more erroneous measurements than good observations. Improvements to the fault detection capability of the method could be obtained by the computation of an inner solution subpaving along with the outer subpaving, or by checking particular points like box centers. This is the main perspective of this research. Finally, this paper proposed a purely static approach, epoch by epoch. A finite memory approach can improve the overall performance by taking a kinematic model into account which allows to use dead-reckoning sensors. The method remains the same but a key stage is, in this case, the management of the data buffer in order to store as much as possible good estimates.

REFERENCES

1. Peyret F, Bétaille D, Mougél F. Non-line-of-sight GNSS signal detection using an on-board 3D model of buildings. *ITS Telecommunications (ITST), 2011 11th International Conference on*, 2011; 280–286, doi: 10.1109/ITST.2011.6060069.
2. Bétaille D, Peyret F, Ortiz M, Miquel S, Fontenay L. A new modeling based on urban trenches to improve GNSS positioning quality of service in cities. *Intelligent Transportation Systems Magazine* 2013; **5**(3):59–70.
3. Walter T, Enge P. Weighted RAIM for precision approach. *Proceedings of ION GPS*, vol. 8, Institute of Navigation, 1995; 1995–2004.
4. Ding S. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Verlag, 2008.
5. Marx B, Maquin D, J R. State estimation and diagnosis of uncertain system based on an interval approach. *Conference on Control and Fault-Tolerant Systems, SysTol'10*, Nice, France, 2010.
6. Raïssi T, Videau G, Zolghadri A. Interval observer design for consistency checks of nonlinear continuous-time systems. *Automatica* 2010; **46**(3):518 – 527, doi:10.1016/j.automatica.2009.12.005.
7. Adrot O, Janati-Idrissi H, Maquin D. Fault detection based on interval analysis. *15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 2002.
8. Fagarasan I, Ploix S, Gentil S. Causal fault detection and isolation based on a set-membership approach. *Automatica* 2004; **40**:2099–2110.
9. Reppa V, Tzes A. Fault detection based on orthotopic set membership identification for robot manipulators. *17th IFAC World Congress (IFAC'08)*, Seoul, Korea, 2008; 7344–7349.
10. Drevelle V, Bonnifait P. A set-membership approach for high integrity height-aided satellite positioning. *GPS Solutions* 2011; **15**(4):357–368, doi:10.1007/s10291-010-0195-3.
11. Schön S, Kutterer H. Uncertainty in GPS networks due to remaining systematic errors: the interval approach. *Journal of Geodesy* 2006; **80**(3):150–162.
12. Ingimundarson A, Bravo JM, Puig V, Alamo T, Guerra P. Robust fault detection using zonotope-based set-membership consistency test. *International Journal of Adaptive Control and Signal Processing* April 2009; **23**(4):311–330.
13. Bai EW, Cho H, Tempo R, Ye Y. Optimization with few violated constraints for linear bounded error parameter estimation. *IEEE Transactions on Automatic Control* July 2002; **47**(7):1067–1077.
14. Leger JB, Kieffer M. Guaranteed robust distributed estimation in a network of sensors. *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010; 3378–3381, doi: 10.1109/ICASSP.2010.5496002.
15. Drevelle V, Bonnifait P. Localization confidence domains via set inversion on short-term trajectory. *IEEE Transactions on Robotics* October 2013; **29**(5):1244–1256.
16. Moore R. *Interval analysis*. Prentice Hall, 1966.
17. Jaulin L, Kieffer M, Didrit O, Walter É. *Applied Interval Analysis*. Springer-Verlag, 2001.
18. Benhamou F, Goualard F, Granvilliers L, Puget JF. Revising hull and box consistency. *Int. Conf. on Logic Programming*, MIT press, 1999; 230–244.
19. Jaulin L, Walter É, Didrit O. Guaranteed robust nonlinear parameter bounding. *CESA'96 IMACS Multiconference (Symposium on Modelling, Analysis and Simulation)*, vol. 2, Lille, 1996; 1156–1161.

20. Le Marchand O, Bonnifait P, Ibañez-Guzmán J, Betaille D, Peyret F. Characterization of GPS multipath for passenger vehicles across urban environments. *ATTI dell'Istituto Italiano di Navigazione* Jul 2009; (189):77–88.

A. COMPUTATION OF THE CENTER OF GRAVITY

With the same idea to prepare the subpaving analysis during the set inversion process, it is possible to apply the same strategy for the computation of the center of gravity. Indeed, the center of gravity is often use as a point position estimate. The following algorithm is designed to compute the subpaving's center of gravity and the compatibility of each measurement with the solution during the set inversion process.

Algorithm 4 RSIVIAExtCoG(in: $[\mathbf{x}], \mathcal{C}_{\{q\}}, \varepsilon$; out: $\mathbf{x}_{cog}, \overline{\mathbb{S}}_x$)

```

 $\mathcal{L} := \{[\mathbf{x}]\}$ 
 $\overline{\mathbb{S}}_x := \emptyset$ 
// Initialize the center-of-gravity (cog) accumulator
 $v := \text{volume}([\mathbf{x}])$ 
 $\mathbf{x}_{cog} := v \cdot \text{mid} [\mathbf{x}]$ 
while  $\mathcal{L} \neq \emptyset$  and not timeout {
    // Pick a box from the list
     $[\mathbf{x}] := \text{pull}(\mathcal{L})$ 
    // Remove the current box from the cog accumulator
     $v := v - \text{volume}([\mathbf{x}])$ 
     $\mathbf{x}_{cog} := \mathbf{x}_{cog} - \text{volume}([\mathbf{x}]) \cdot \text{mid} [\mathbf{x}]$ 
    // q-relaxed contraction with measurements
     $[\mathbf{x}] := \mathcal{C}_{\{q\}}([\mathbf{x}])$ 
    if  $[\mathbf{x}] \neq \emptyset$  then {
        // Update the cog accumulator
         $v := v + \text{volume}([\mathbf{x}])$ 
         $\mathbf{x}_{cog} := \mathbf{x}_{cog} + \text{volume}([\mathbf{x}]) \cdot \text{mid} [\mathbf{x}]$ 
        if  $w([\mathbf{x}]) < \varepsilon$  then { // Box is too small
             $\overline{\mathbb{S}}_x := \overline{\mathbb{S}}_x \cup [\mathbf{x}]$ 
        }
        else { // Bisect and enqueue subboxes
             $([\mathbf{x}_1], [\mathbf{x}_2]) = \text{bisect}([\mathbf{x}])$ 
             $\text{push}([\mathbf{x}_1], \mathcal{L}); \text{push}([\mathbf{x}_2], \mathcal{L})$ 
        }
    }
}
// Compute the solution set
 $\overline{\mathbb{S}}_x := \overline{\mathbb{S}}_x \cup \mathcal{L}$ 
// Compute the center of gravity
 $\mathbf{x}_{cog} := \mathbf{x}_{cog} / v$ 

```
