The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Robust positioning using relaxed constraint-propagation

Vincent Drevelle and Philippe Bonnifait

Abstract— **A robust 2-D vehicle-positioning system based on constraint propagation on a data horizon is proposed. Using asynchronous unreliable absolute positions and reliable proprioceptive measurements, the proposed method outputs the bounds of the pose at any desired rate. A bounded error approach is used to propagate the imprecision of the input data to the final pose estimation. Measurements are represented by intervals taking errors into account. Integrity is assumed on proprioceptive measurements error bounds. In opposition, the bounds set on exteroceptive positions are not required to be guaranteed. Indeed, a relaxed intersection of constraints is applied to achieve robustness, given a maximum ratio of erroneous positions. This approach is implemented for an automotive vehicle equipped with a GPS receiver and a CAN gateway. An real experimental validation is carried out with a dataset including wrong GPS measurements and the crossing of a tunnel.**

Index Terms— **localization, interval analysis, multi-sensor fusion**

I. Introduction

Robotic systems often use exteroceptive sensors to get absolute position information. GPS or radio beacons [1] can be used for this purpose. This absolute position information is however not always available due to masking, and may also suffer from aberrant measurements (multipath propagation, interferences, etc). Moreover, the sampling rate of those absolute positioning sensors is often too slow for control tasks, and attitude information is unavailable at slow speeds. Proprioception is used to get relative motion information, by the means of wheel speed encoders and inertial sensors (gyrometers and accelerometers). Those sensors operate at a high sampling rate, but an initial absolute pose is mandatory to integrate the measurements in a dead-reckoning approach. Furthermore, measurement errors accumulate at each integration step, which results in drift for long integration times. Proper fusion between absolute and proprioceptive sensor data enables to get high frequency absolute pose estimation, by cancelling the integration drift of proprioceptive sensors. Pose inaccuracy, which is a key information for a decision task, also has to be estimated.

Recursive filtering has been the preferred solution for state estimation for long time, mainly because of its very low memory requirements: no need to keep track of previous measurement or estimated states, only the current state is stored since the state is an information vector which enables one-step ahead prediction. This makes this estimation scheme suitable for real-time implementation in low-memory embedded systems. Advances in technology have raised the available memory and computing power of embedded systems, enabling to keep more information in memory for processing. This gives the means to buffer an history of past measurements and inputs, and to handle trajectories.

Processing buffered inputs and measurements as several advantages over the recursive filtering scheme:

- Out Of Sequence measurements (sensors latency and asynchronous data) can be managed by adding data at their acquisition date in the buffer, while it is very cumbersome to add past information after an update of a recursive filter.
- Multiple or persistent faults can be detected. Indeed, while one-state memory can only predict the current output, the buffered data can be exploited for several epochs.
- Data association can be enhanced, like in the lazy data association algorithm of [2] which allows revising association in the past when a wrong association is detected.

Several methods have been used to process buffered measurements. A quadratic optimisation scheme can be used to fit a trajectory to the data [3]. Multi-update filtering [4] and fixed-lag filtering [5] allows taking into account late measurements of high latency sensors even if other measurements have been gathered in the meantime. Set-inversion and constraint propagation methods on intervals have also been used to provide guaranteed estimation using bounded error state estimation [6], [7]. A guaranteed loosely-coupled fusion over a data horizon method has been proposed in [8], but it cannot handle the presence of outliers. In [9], a robust set membership method is proposed to localize a submarine with a buffer of unreliable sonar measurements.

Using a robust interval based solver, we compute the 2-D pose of a vehicle given a finite history of absolute position measurements and proprioceptive data. Our method handles out-of-sequence measurements with a data horizon, and a q-relaxed approach enables to deal with erroneous positions. As long as the ratio of wrong position measurements in the buffer is below a defined value, the computed poses can be guaranteed.

The paper is organized as follows. After an overview of the proposed pose estimation system and data horizon management, interval analysis and robust set-inversion are briefly introduced. Then, the system implementation is described, with a simple vehicle model, a set-inversion method to compute bounded-error GPS positions, and the robust pose estimator based on a horizon of proprioceptive data

V. Drevelle and P. Bonnifait are with Heudiasyc, CNRS UMR 6599, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, BP 20529, 60205 Compiegne Cedex, France
vincent.drevelle@hds.utc.fr

and absolute positions. Finally, experimental results with real GPS and odometry are presented.

II. Problem Statement

A. Pose estimation system overview

The proposed system computes the pose of a vehicle equipped with an absolute position sensor and proprioceptive sensors. It combines low sample rate position measurements with high acquisition rate proprioceptive data, over a data horizon (Fig. 1). Pose estimation has to be output at any desired time. This problem is of general relevance in robotics: the exteroceptive sensor can be any absolute system able to compute a location (a GPS receiver, a camera, a radio system, a lidar, etc.).

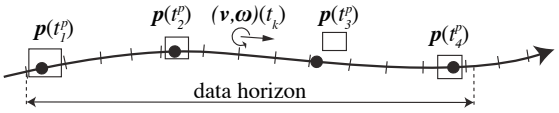


Fig. 1: Data horizon. *Ticks on the trajectory represent proprioceptive measurements, while circles represent position measurements epochs. The boxes show the positions input to the system. Notice an erroneous measurement at time* t_3^p.

Fig. 2 shows an overview of the fusion system. A first subsystem provides bounded-error absolute positions. It can be the computation of a GPS position from raw pseudorange measurements. Positions and proprioceptive data are then merged using a robust algorithm based on interval analysis and a buffer of past measurements and inputs. A data buffer management algorithm supervises buffers filling, keeping a reasonable buffer size and ensuring buffers hold enough information to estimate the pose.

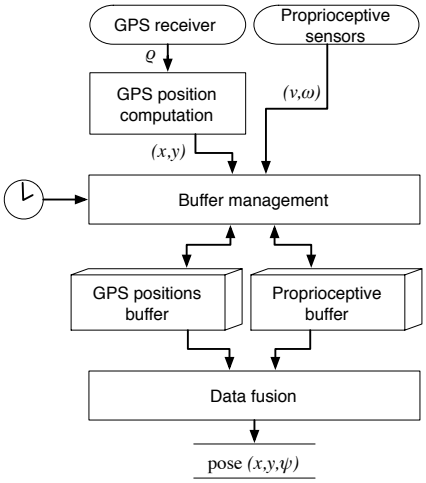


Fig. 2: Data fusion system overview

Data is represented as intervals and boxes throughout the system. This allows propagating errors in a guaranteed way, from the measurements to the pose estimation. To model inaccuracy, positions are represented as boxes which should contain the true location with a given confidence level. Measurements are represented by intervals to take noise into account.

B. Data history buffers

To allow pose estimation based on a finite number of past observations, two data history buffers are used:

- the list of absolute position observations boxes, containing o positions: $L_p(t) = \{\boldsymbol{p}(t_1^p), \ldots, \boldsymbol{p}(t_o^p)\}$
- the list of proprioceptive inputs boxes, containing n boxes: $L_u(t) = \{\boldsymbol{u}(t_1^u), \ldots, \boldsymbol{u}(t_n^u)\}$, with $\boldsymbol{u}(t) = (\boldsymbol{v}(t), \boldsymbol{\omega}(t))^T$, $v(t)$ and $\omega(t)$ denoting respectively the linear speed and the angular speed of the vehicle at the midpoint of the rear axle

Each record in the data history buffer is dated with its time of acquisition, to allow variable acquisition rate processing.

Data history buffers are managed to keep a tractable size. The growth of the list of position observations is limited. When the size limit is reached, the oldest data is removed to make room for incoming data. The list of proprioceptive inputs is then cleaned from obsolete data related to the position previously discarded.

Since the absolute positions buffer is of limited size, adding new observation data implies forgetting older position data. Adding every position to the observation buffer may lead to heading estimation issues when the length of the buffered trajectory is in the same order of magnitude as the position boxes width. This problem arises when the vehicle stops: the system starts to accumulate redundant position observations, while discarding older parts of the trajectory and thus loosing the constraint on heading.

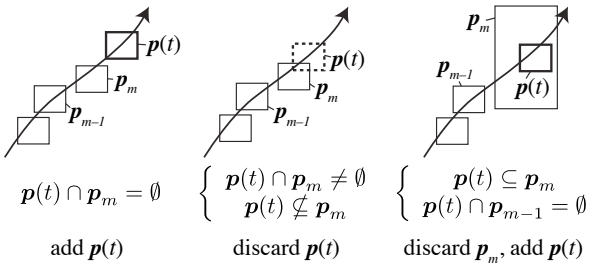


Fig. 3: Data horizon management policy

To address this issue, observation buffer filling is based on a spatial criterion (Fig. 3). If the new position box to be added intersects the last position in the buffer, it is not added to the buffer. This rule can however lead to the loss of informative position information when the last position box is too large, preventing any new smaller (thus more informative) box to be added. To counteract this side-effect, the buffer filling policy is complemented by a second rule: if the box to be added is included in the last box of the buffer, and if it does not intersect the penultimate box of the buffer, then the buffer's last box is replaced by the new box.

III. Interval Analysis and Robust Set Inversion

A. Interval analysis

Interval analysis [10] involves intervals and their multidimensional extension, *interval vectors* (or *boxes*). In opposition to an exact representation of sets, intervals and boxes are easy to represent and manipulate. The set of real intervals is denoted $\mathbb{IR}$, and the set of n-dimensional boxes is $\mathbb{IR}^n$. In this paper, an interval or a box $\boldsymbol{x} = [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ is written in bold; $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ respectively denote the lower and upper bounds of $\boldsymbol{x}$.

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

$$\boldsymbol{x} + \boldsymbol{y} = [\underline{\boldsymbol{x}} + \underline{\boldsymbol{y}}, \overline{\boldsymbol{x}} + \overline{\boldsymbol{y}}], \quad \boldsymbol{x} - \boldsymbol{y} = [\underline{\boldsymbol{x}} - \overline{\boldsymbol{y}}, \overline{\boldsymbol{x}} - \underline{\boldsymbol{y}}]$$

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 $f : \mathbb{R}^n \to \mathbb{R}^m$ 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 $\boldsymbol{x}$ by f is included in the image of $\boldsymbol{x}$ by f, *i.e.*

$$\forall \boldsymbol{x} \in \mathbb{IR}^n, f(\boldsymbol{x}) \subset \mathsf{f}(\boldsymbol{x}).$$

The minimal inclusion function f^* for a function f returns the smallest box that contains $f(\boldsymbol{x})$ — *i.e.* $\mathsf{f}^*(\boldsymbol{x}) = \square f(\boldsymbol{x})$, the *interval hull* of $f(\boldsymbol{x})$ [10].

To approximate compact sets in a guaranteed way, *subpavings* can be used. A subpaving of a box $\boldsymbol{x}$ is the union of non-empty and non-overlapping subboxes of $\boldsymbol{x}$. A guaranteed approximation of a compact set X can be made by bracketing it between an inner subpaving X_{in} and an outer subpaving X_{out} such as $X_{in} \subseteq X \subseteq X_{out}$.

B. Robust set inversion

The set inversion problem consists in determining the set X such as $f(X) = \boldsymbol{y}$, where $\boldsymbol{y}$ is a known interval vector of m measurements. Using interval analysis, the solution $X = f^{-1}(\boldsymbol{y})$ can be approximated between two subpavings X_{in} and X_{out} such that $X_{in} \subseteq X \subseteq X_{out}$. The *SIVIA* algorithm allows performing such a set inversion, by recursively bisecting an initial box [10].

When the measurements interval vector $\boldsymbol{y}$ may contain erroneous data, a *robust set inversion* can be performed. A robust set inversion method is the *q-relaxed set inversion*. To perform a q-relaxed set inversion with a m-dimensional measurements vector $\boldsymbol{y}$, a given number q of wrong measurements is tolerated. The solution set is thus the set of solutions *at least compatible with $m - q$ measurements*.

Considering m sets $X_1, \ldots, X_m$ of $\mathbb{R}^n$, the *q-relaxed intersection* $\bigcap^{\{q\}} X_i$ is the set of $x \in \mathbb{R}^n$ which belongs to at least $m - q$ of the X_i's (Fig. 4).

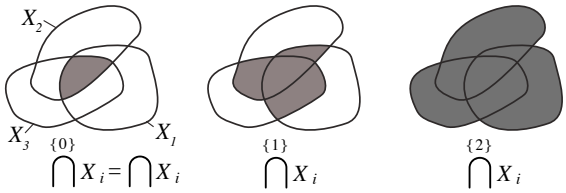


Fig. 4: q-relaxed intersection of three sets for $q \in \{0, 1, 2\}$

By considering $X_i = f_i^{-1}(\boldsymbol{y_i})$, the *Robust Set Inverter via Interval Analysis* (RSIVIA) solver [10] guarantees the computation of a q-relaxed solution set for $X = f^{-1}(\boldsymbol{y})$ (see Algorithm 1). This algorithm returns an outer subpaving of the q-relaxed solution.

Algorithm 1 RSIVIA(in: $\boldsymbol{x_0}$, f, $\boldsymbol{y}$, q ; out: X)
Robust Set Inverter via Interval Analysis

1: $\text{push}(\boldsymbol{x_0}, \mathfrak{L})$
2: **while** $\mathfrak{L} \neq \emptyset$ **do**
3: $\quad \boldsymbol{x} = \text{pull}(\mathfrak{L})$
4: $\quad$ **repeat**
5: $\quad\quad$ **for** $i = 1 \ldots m$ **do**
6: $\quad\quad\quad$ compute $\boldsymbol{x}(i)$ enclosing $\boldsymbol{x} \cap f_i^{-1}(\boldsymbol{y}_i)$
7: $\quad\quad$ **end for**
8: $\quad\quad \boldsymbol{x} = \square \left(\bigcap_{i \in \{1,\ldots,m\}}^{\{q\}} \boldsymbol{x}(i) \right)$ *hull of the q-relaxed intersection of m boxes*
9: $\quad$ **until** no more contraction can be done on $\boldsymbol{x}$
10: $\quad$ **if** $\boldsymbol{x} \neq \emptyset$ **then**
11: $\quad\quad (\boldsymbol{x_1}, \boldsymbol{x_2}) = \text{bisect}(\boldsymbol{x})$
12: $\quad\quad \text{push}(\boldsymbol{x_1}, \mathfrak{L})$; $\text{push}(\boldsymbol{x_2}, \mathfrak{L})$
13: $\quad$ **end if**
14: **end while**
15: **return** $\mathfrak{L}$

In this solver, the computation of a bounding box of $\boldsymbol{x} \cap f_i^{-1}(\boldsymbol{y}_i)$ is generally performed using interval constraint propagation methods, like the *Fall-Climb* algorithm [10].

If a measurement is wrong and inconsistent with the other measurements, it is automatically excluded from the solution, and it can be identified as an outlier.

IV. Implementation

A. Vehicle model

The vehicle is assumed to move without slipping in an horizontal planar world. The continuous-time evolution model of the vehicle is

$$\begin{cases} \dot{x}(t) = & v(t) \cdot \cos\psi(t) \\ \dot{y}(t) = & v(t) \cdot \sin\psi(t) \\ \dot{\psi}(t) = & \omega(t) \end{cases}$$

After discretization the model becomes (T_s being the sampling period)

$$\begin{cases} x_{k+1} = & x_k + T_s \cdot v_k \cdot \cos\psi_k \\ y_{k+1} = & y_k + T_s \cdot v_k \cdot \sin\psi_k \\ \psi_{k+1} = & \psi_k + T_s \cdot \omega_k \end{cases}$$

The vehicle linear speed and angular speed are usually measured using differential odometry. $\boldsymbol{v_R}(t)$ and $\boldsymbol{v_L}(t)$ are respectively the right and left wheel speed. $\boldsymbol{L}$ is the distance between the two wheels. These quantities are intervals, whose bounds are set to represent the measurement error.

$$\begin{cases} \boldsymbol{v}(t) = & \frac{\boldsymbol{v_R}(t)+\boldsymbol{v_L}(t)}{2} \\ \boldsymbol{\omega}(t) = & \frac{\boldsymbol{v_R}(t)-\boldsymbol{v_L}(t)}{\boldsymbol{L}} \end{cases}$$

If the vehicle is equipped with a gyrometer, $\omega(t)$ can be directly measured.

B. Bounded-error GPS position

To be used in the 2-D robust pose-estimation, absolute position boxes have to be in local frame coordinates. Position estimated by the GPS receiver could be transformed into local frame coordinates, and a bounding box could be computed using the position variance information computed by the receiver. However, setting bounds in local frame on a position computed in a global frame is not trivial.

We use an interval method to compute position boxes, by setting bounded errors on GPS pseudo-range measurements [11]. Computation is directly done in local frame coordinates, so that solution boxes are already aligned with the evolution plane of the vehicle. This way, error on measurements are already propagated in the appropriate coordinates frame.

GPS positioning is a *Time of Arrival* method, which involves *pseudorange* measurements to each of the visible satellites [12]. Pseudoranges are offset by a unknown amount due to the time base difference between the receiver and the GPS system. GPS positioning using pseudoranges is thus a four-dimensional problem: along with the Cartesian coordinates (x, y, z) of the user, the user's clock offset dt_u has to be estimated. Satellite positions (x_i^s, y_i^s, z_i^s) are computed with broadcast ephemeris data. They are represented as intervals $(\boldsymbol{x_i^s}, \boldsymbol{y_i^s}, \boldsymbol{z_i^s})$ to take ephemeris inaccuracy into account. The propagation delay corrections applied to measured pseudoranges to get corrected pseudoranges ρ_i are imprecise because of model and parameter errors. Moreover, the receiver measurement errors should also be taken into account. Corrected pseudorange measurements are thus represented as intervals $\boldsymbol{\rho_i}$ whose bounds are determined given an integrity risk [11].

The location zone computation consists in 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 induced by the i^{th} pseudorange measurement is represented by the natural inclusion function of the GPS pseudorange observation function:

$$\mathsf{f_i}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{dt_u}) = \sqrt{(\boldsymbol{x}-\boldsymbol{x_i^s})^2 + (\boldsymbol{y}-\boldsymbol{y_i^s})^2 + (\boldsymbol{z}-\boldsymbol{z_i^s})^2} + c \cdot \boldsymbol{dt_u} \quad (1)$$

The *Fall-Climb* algorithm [13] allows constraints to be propagated in an optimal order for each measurement, using (1). Position computation is then performed using the *RSIVIA* algorithm presented previously (Alg. 1). It allows computing a position robust to a given number of erroneous pseudorange measurements. For further details, please see [11].

C. Robust pose estimation using the buffers

Two dated data history buffers are used: the list of position observations boxes $L_p(t) = \{\boldsymbol{p}(t_1^p), \ldots, \boldsymbol{p}(t_o^p)\}$, and the list of proprioceptive inputs boxes L_u(t)=$\{\boldsymbol{u}(t_1^u), \ldots, \boldsymbol{u}(t_n^u)\}$, with $\boldsymbol{u}(t) = (\boldsymbol{v}(t), \boldsymbol{\omega}(t))^T$.

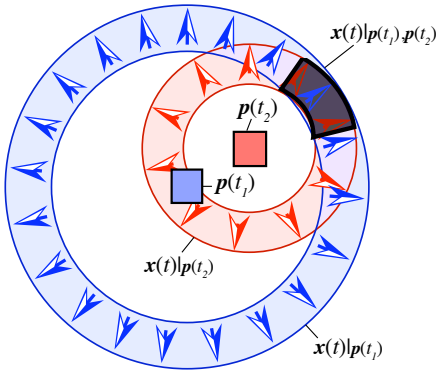


Fig. 5: Estimation of current pose, given two absolute positions

Fig. 5 shows the estimation of the current pose at time t, given two positions at times t_1 and t_2, and proprioceptive data history. Each position constrains current position and attitude in a compact set of the pose space (projected as a ring-like shape in 2-D view of Fig. 5). The current pose is found at the intersection of the constraints imposed by each position information (2).

$$\boldsymbol{x}(t) = \bigcap_{k=1\ldots o} \boldsymbol{x}(t)|\boldsymbol{p}(t_k^p), \boldsymbol{u}(t_1^u), \ldots, \boldsymbol{u}(t_n^u) \quad (2)$$

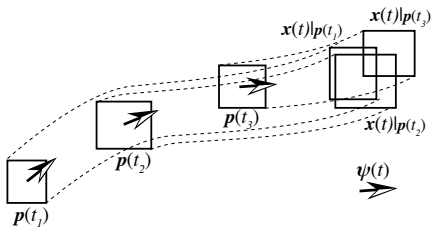
To deal with erroneous positions in the data buffer, a given number q of erroneous measurements in the buffer is tolerated, using a q-relaxed intersection (3). The robust pose estimation consists in computing the vehicle's position and heading at time t, given a finite number of prior position measurements and the history of inputs (proprioceptive sensors), under the hypothesis that at most q position measurements are wrong.

$$\boldsymbol{x}(t) = \bigcap_{k=1\ldots o}^{\{q\}} \boldsymbol{x}(t)|\boldsymbol{p}(t_k^p), \boldsymbol{u}(t_1^u), \ldots, \boldsymbol{u}(t_n^u) \quad (3)$$

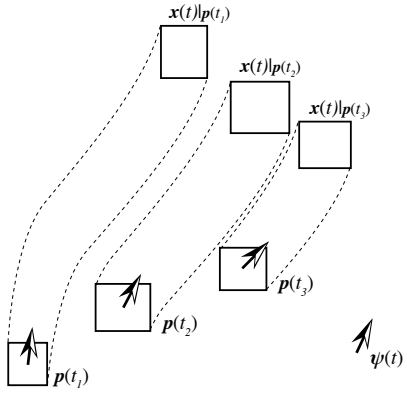
Computation of $\boldsymbol{x}(t)|\boldsymbol{p}(t_k^p), \boldsymbol{u}(t_1^u), \ldots, \boldsymbol{u}(t_n^u)$ is done using a backward-forward constraint propagation [10] with the evolution function and the proprioceptive inputs. Since each $\boldsymbol{p}(t_k^p)$ only contains position information, $\boldsymbol{\psi}(t)$ cannot be estimated directly by independent contraction with respect to each $\boldsymbol{p}(t_k^p)$. To address this issue, the initial range of $\boldsymbol{\psi}(t) \subseteq [0, 2\pi]$ is partitioned in N intervals $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N\}$, and constraint propagation is performed for each $\boldsymbol{\psi}_k$ value (4), see Fig. 6. Algorithm 2 summarizes the whole computation.

$$\boldsymbol{x}(t) = \bigcup_{k=1}^{N} \boldsymbol{x}(t)|\psi(t) \in \boldsymbol{\psi}_k, \text{ with } \bigcup_{k=1}^{N} \boldsymbol{\psi}_k = [0, 2\pi] \quad (4)$$

Algorithm 2 is a simplified RSIVIA algorithm. The main difference is that it only performs a partitioning of the search space among $\boldsymbol{\psi}(t)$, while RSIVIA would perform bisections along all dimensions of the search space.



(a) Prior $\psi(t)$ interval is compatible with measurements history



(b) Prior $\psi(t)$ interval is inconsistent with measurements history, the intersection of estimated poses is empty

Fig. 6: Results of backward-forward constraint propagation with three position measurements, for two different prior values of $\psi(t)$. Arrows represent the heading.

Algorithm 2 Compute current pose

$\boldsymbol{x}(t) \leftarrow \emptyset$
for $k = 1 \dots N$ **do**
  $\boldsymbol{\psi}_k = [\frac{2\pi(k-1)}{N}, \frac{2\pi k}{N}]$
  for $i = 1 \dots o$ **do**
    $\boldsymbol{\psi}(t_i^p) = \text{bwd\_propag}(\boldsymbol{\psi}_k)$
    $\boldsymbol{x}(t)|\boldsymbol{p}(t_i^p), \psi(t) \in \boldsymbol{\psi}_k = \text{fwd\_propag}\begin{pmatrix} \boldsymbol{p}(t_i^p) \\ \boldsymbol{\psi}(t_i^p) \end{pmatrix}$
  end for
  $\boldsymbol{x}(t)|\psi(t) \in \boldsymbol{\psi}_k = \bigcap_{i=1\dots o}^{\{q\}} \boldsymbol{x}(t)|\boldsymbol{p}(t_i^p), \psi(t) \in \boldsymbol{\psi}_k$
  $\boldsymbol{x}(t) \leftarrow \boldsymbol{x}(t) \cup \boldsymbol{x}(t)|\psi(t) \in \boldsymbol{\psi}_k$
end for

D. Proprioceptive data integration

Proprioceptive data integration is required for constraint propagation in Alg. 2. The *bwd_propag* step computes the vehicle's heading at a past epoch, given the current heading and the history of measurements. The *fwd_propag* function computes the current pose, given a pose in the past and the proprioceptive measurements.

Integration of proprioceptive inputs using interval analysis can be very conservative, since multiple occurrences of the same variable are considered independently. To address this issue, monotonicity can be exploited. If f is increasing in $\boldsymbol{x_i}$, decreasing in $\boldsymbol{x_d}$, and non-monotonic on $\boldsymbol{z}$, a better evaluation of the the range of f over $(\boldsymbol{x_i}, \boldsymbol{x_d}, \boldsymbol{z})$ than its natural expression f is given by $\mathsf{f_m}$ (5)

$$\mathsf{f_m}(\boldsymbol{x_i}, \boldsymbol{x_d}, \boldsymbol{z}) = [\underline{\mathsf{f}(\underline{\boldsymbol{x_i}}, \overline{\boldsymbol{x_d}}, \underline{\boldsymbol{z}})}, \overline{\mathsf{f}(\overline{\boldsymbol{x_i}}, \underline{\boldsymbol{x_d}}, \overline{\boldsymbol{z}})}] \tag{5}$$

This property is used to perform integration of proprioceptive data with less pessimism when monotonicity of the evolution function can be proved over the range of the current state and observations.

When the absolute positions list has not changed since the last estimation $\boldsymbol{x}(t_{last})$, and providing that $\boldsymbol{x}(t_{last})$ and the new $\boldsymbol{x}(t)$ to be estimated are posterior to the last position observation $\boldsymbol{p}(t_m^y)$, $\boldsymbol{x}(t)$ can be computed using only proprioceptive input integration from $\boldsymbol{x}(t_{last})$. This allows speeding up the computation of the pose between two position measurements.

V. Results

Experimental validation has been performed with an instrumented vehicle in the western suburb of Paris. We recorded GPS pseudorange measurements from a *uBlox* GPS receiver at 1 Hz, and proprioceptive data from the speed sensor and gyrometer of the vehicle at 10 Hz. A high-end *LandINS* inertial navigation system coupled with a high-end *Novatel* GPS receiver was used to provide ground truth.

Fig. 7: Vehicle entering and exiting the tunnel. *On-board camera view.*

The vehicle was driven on a forest-bordered highway, then passed through a 900-meter long tunnel (Fig. 7).

Recorded data is processed in real-time playback. GPS position computation is performed using a non-robust interval GPS solver, to emphasise the fusion process benefits. Fig. 8 shows lots of positions inconsistent with ground truth during the first 140 seconds of highway driving. During this period, the GPS computation sometimes returns no solution due to measurements inconsistency. When the vehicle enters the tunnel from $t = 155\,\text{s}$ until $200\,\text{s}$, the GPS signal is totally blocked, thus preventing position computation.

Output of the robust pose estimation method is shown in Fig. 9. Algorithm is set to tolerate at least 20% of erroneous measurement without loss of integrity. The length of the position buffer is limited to 14 boxes. The system keeps a pose consistent with ground truth during the whole 250 seconds of driving. Despite the misleading GPS positions during the first 140 seconds of highway driving, position bounds in the computed pose are still consistent with ground truth. When no GPS position is available, the computed position box grows, but still remains centered on ground truth.

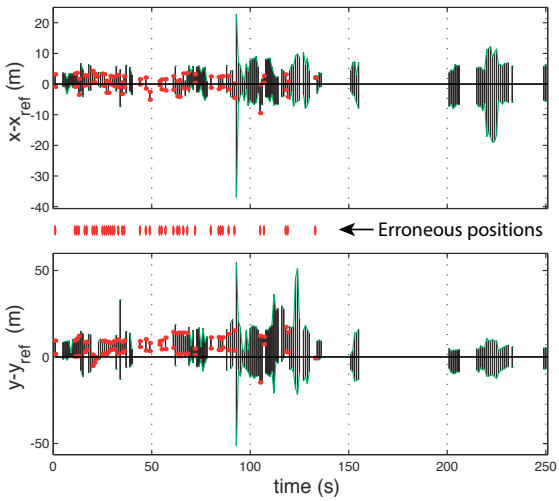


Fig. 8: GPS position boxes input to the fusion system. *Ground truth is zero.*

The increase of position uncertainty during dead-reckoning is the consequence of propagating heading-estimation uncertainty and linear and angular speed measurements uncertainty through the evolution model. The tunnel mainly follows a West-East axis, as a consequence, the increase of uncertainty on the x (West-East) coordinates is mainly due to speed measurements uncertainty. The y coordinates uncertainty grows faster, because it is more dependant of the heading estimation uncertainty.

VI. Conclusion

A method to estimate the 2-D pose of a vehicle using proprioceptive sensors and unreliable absolute positions has been presented in this paper. It uses bounded-error measurements gathered in size-limited data history buffers. Pose estimation is carried out using a robust relaxed constraint-propagation scheme, assuming a defined maximum number of erroneous position measurements in the buffer. Thanks to a position data buffer management based on spatial criteria, loss of information in the position buffer is prevented.

An experimental validation has been performed, using a non-robust GPS solver and proprioceptive measurements (speed and gyro). It showed that the system is able to provide a pose estimation consistent with ground truth, even when misleading position measurements are input. It also demonstrated the system's ability to output consistent pose estimations during long absolute positions unavailability periods.

Future work will be focused on using raw GPS measurements in the data horizon, in a robust tightly-coupled fusion approach. The use of a 3-D representation will also be investigated to deal with positioning in hilly areas.

Acknowledgments

The authors would like to thank C. Fouque, O. Le Marchand and J. Ibañez for their experimental support.

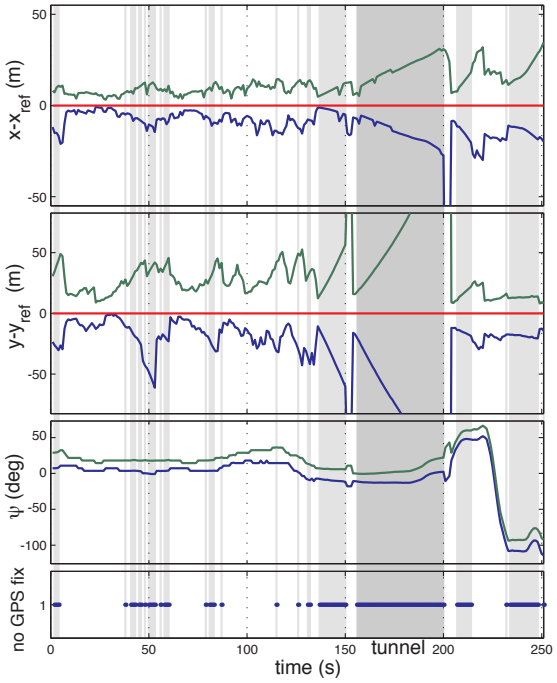


Fig. 9: Results of the robust data fusion. *From top to bottom: Upper and lower bounds of position (zero is ground truth). Upper and lower bounds of absolute heading. Loss of GPS position input (shown in gray).*

References

[1] C. Röhrig and M. Müller, "Indoor location tracking in non-line-of-sight environments using a IEEE 802.15.4a wireless network," in *Conference Proceedings of IROS09. St. Louis*, 10 2009, pp. 552–557.

[2] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, Z. Omohundro *et al.*, "Autonomous exploration and mapping of abandoned mines," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 79–91, 2004.

[3] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, p. 1181, 2006.

[4] C. Tessier, C. Cariou, C. Debain, F. Chausse, R. Chapuis, and C. Rousset, "A real-time, multi-sensor architecture for fusion of delayed observations: application to vehicle localization," in *IEEE ITSC'06*, 2006, pp. 1316–1321.

[5] A. Ranganathan, M. Kaess, and F. Dellaert, "Fast 3D pose estimation with out-of-sequence measurements," in *IROS07*, 2007.

[6] P. Bouron, D. Meizel, and P. Bonnifait, "Set-membership non-linear observers with application to vehicle localisation," in *6th European Control Conference. Porto*, 2001, pp. 1255–1260.

[7] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez, "Consistent outdoor vehicle localization by bounded-error state estimation," in *Conference Proceedings of IROS09. St. Louis*, 2009, pp. 1211–1216.

[8] A. Gning and P. Bonnifait, "Constraints propagation techniques on intervals for a guaranteed localization using redundant data," *Automatica*, vol. 42, no. 7, pp. 1167–1175, 2006.

[9] L. Jaulin, "Robust set-membership state estimation; application to underwater robotics," *Automatica*, vol. 45, no. 1, pp. 202–206, 2009.

[10] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. Springer-Verlag, 2001.

[11] V. Drevelle and P. Bonnifait, "High integrity GNSS location zone characterization using interval analysis," in *Proceedings of ION GNSS 2009*, 2009, pp. 2178–2187.

[12] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications Second Edition*. Artech House Publishers, 2006.

[13] L. Jaulin, M. Kieffer, I. Braems, and E. Walter, "Guaranteed nonlinear estimation using constraint propagation on sets," *Int. J. Control*, vol. 74, no. 18, pp. 1772–1782, 2001.