Proving the existence of loops in robot trajectories

Simon Rohou, Peter Franek, Clément Aubry, Luc Jaulin

Proving the existence of loops in robot trajectories

Simon Rohou[a], Peter Franek[b], Clément Aubry[c], Luc Jaulin[a]

[a] *ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France*
[b] *IST Austria*
[c] *ISEN Brest, France*

Abstract

This paper presents a reliable method to verify the existence of loops along the uncertain trajectory of a robot, based on proprioceptive measurements only, within a bounded-error context. The loop closure detection is one of the key points in SLAM methods, especially in homogeneous environments with difficult scenes recognitions. The proposed approach is generic and could be coupled with conventional SLAM algorithms to reliably reduce their computing burden, thus improving the localization and mapping processes in the most challenging environments such as unexplored underwater extents. To prove that a robot performed a loop whatever the uncertainties in its evolution, we employ the notion of *topological degree* that originates in the field of differential topology. We show that a verification tool based on the topological degree is an optimal method for proving robot loops. This is demonstrated both on datasets from real missions involving autonomous underwater vehicles, and by a mathematical discussion.

Keywords: mobile robotics, SLAM, loop detection, interval analysis, topological degree, tubes

1. Introduction

The SLAM, Simultaneous Localization And Mapping [1, 2], is an approach that ties together the problem of state estimation and the one of mapping an unknown environment. Basically, a robot coming back to a previous pose is likely to recognize an old scene and then refine its localization. The key point of these methods is then to detect that a place has been previously visited. This problem of data association is known in the literature as loop closure [3].

1.1. Detecting loop closures

A loop can be detected thanks to *exteroceptive* measurements, *i.e.* the perception of the outside, by scenes comparisons [4, 5, 6, 7]. However, it can be difficult to detect the closure due to poor estimations on both the robot's position and map-matchings. The problem appears even more challenging when dealing with homogeneous environments without any point of interest to rely on. This is typically the case one can encounter in underwater exploration with wide homogeneous sea-floors. Such situation will unfortunately lead to a few detections of confident loop closures or, in the worst cases, to false detections that could lead to a wrong localization and mapping.

Besides exteroceptive measurements, it has been shown in [8] that loops can be approximated based on *proprioceptive* measurements only, namely: velocity vectors and inertial values knowing the kinematic of the robot. This approach has the advantage to be applicable regardless of the nature of the environment to explore. Of course, one should note that in this case, the loop detections cannot improve by themselves the localization, as the approach will not bring new exteroceptive information and so no new constraints to the problem.

However, this method is of high interest if combined with classical SLAM techniques that merge both proprioceptive and exteroceptive measurements, in order to decrease the computing burden of usual scenes recognitions. Indeed, the complexity of SLAM algorithms quickly increases with the exploration of wide environments, as it implies lots of loop closures to identify among a dense set of data. To this day, the execution of SLAM programs in 3D environments during long-term missions is often not affordable for classical embedded systems powering the robots. A part of the community hence focuses on light-weight solutions. This work is heading in this direction, proposing a way to estimate the loop closures that does not rely on environment observations. This approach is then guaranteed to provide real-time results as it does not go into a costly analysis of observation datasets.

On top of that, a reliable approach that provides guaranteed loop approximations is suited to prevent from false detections in singular environments. This situation is typically encountered when two different objects of same properties are considered as unique by algorithms standing on

too uncertain positioning estimations. Figure 1 gives an example of same looking objects and uncertain trajectories estimations. This situation may lead to the detection of wrong loop closures. Our method provides a way to reject the feasibility of a loop closure despite the ambiguity of the situation.

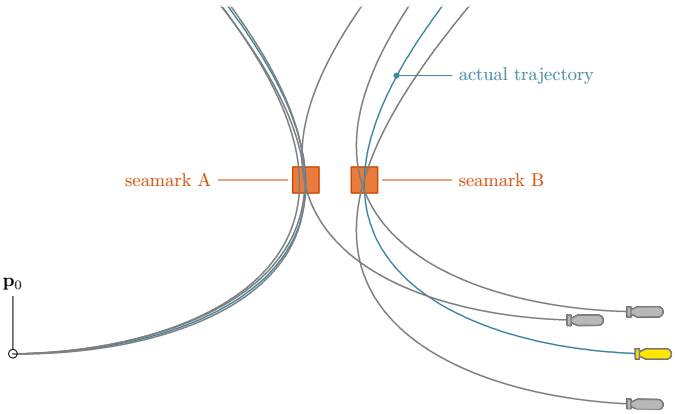


Figure 1: A robot flying over two different but same-looking seamarks. The actual trajectory is plotted in blue while several dead-reckoning estimations are drawn in gray. All the trajectories are consistent with the observations. A well-known map would not prevent from wrong loop closure detections.

1.2. The two-dimensional case

Formally, a robot that performed a loop is a robot that came back to a previous position $\mathbf{p}(t)$. The main contribution of this paper is to provide a reliable method to prove the existence of loops along the uncertain trajectory of a robot, based on proprioceptive measurements only, using a bounded-error approach. In a reliable context, a distinction has to be made between the *detection* and the *verification* (*i.e.* the proof) of a loop. Considering a set of feasible trajectories as the one displayed in Fig. 2, some of them may cross themselves at some point; this will lead to a *detection*. In addition, when we verify that all the feasible trajectories are looped, then we can speak about a *loop proof* since a loop occurs whatever the considered uncertainties.

We will focus on loops along two-dimensional trajectories: $\mathbf{p}(t) \in \mathbb{R}^2$. This choice is not a limitation made to keep things simple, but a practical requirement. Indeed, it is not possible to physically verify $\mathbf{p}(t_1) = \mathbf{p}(t_2)$ in higher dimensional spaces. A robot will never reach again the very same 3D atomic position, in contrast with two-dimensional cases. Furthermore, the amount of uncertainties we have to deal with will always be too large to verify this. Therefore, it is not possible to prove three-dimensional loops, nor to verify that a robot came back to a previous *pose*, including both position and orientation, for the same reason.

Verify a two-dimensional loop is still interesting for many

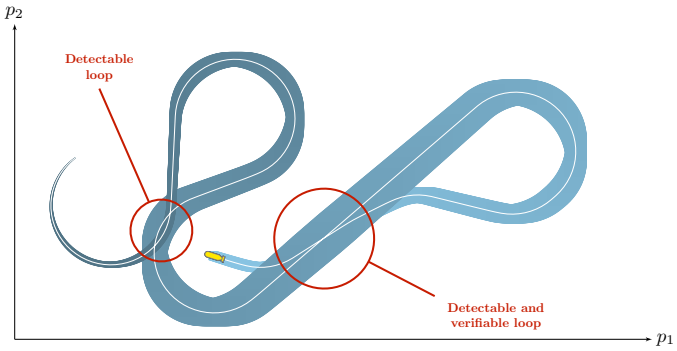


Figure 2: This view represents the uncertain evolution of a mobile robot. Feasible trajectories are enclosed within a *tube*, displayed in blue. Here, only one loop can be *verified*, while at least two feasible loops are *detectable*. Indeed the actual trajectory, plotted in white, loops three times whereas there exist trajectories in the tube that loop only once.

3D applications. For instance, as pictured in Figure 3, an underwater robot can apply a raw-data SLAM method using a sonar for exteroceptive measurements. In this configuration, the SLAM can be reduced to a 2D problem by merging vertical measurements, namely: depth from a pressure sensor and altitude from the sonar. Map-matching will then be achievable over each 2D crossing, as pictured in the figure with projections on the seafloor.

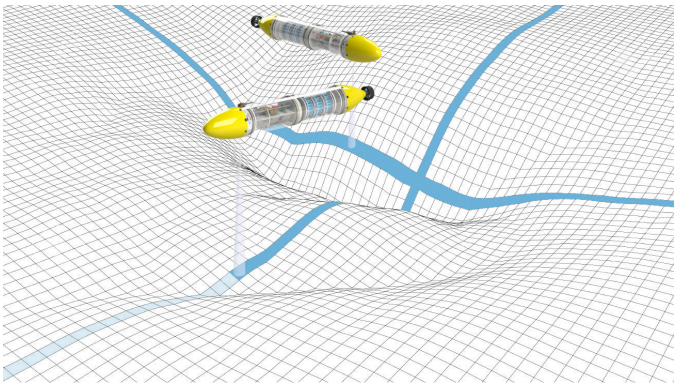


Figure 3: An underwater robot exploring its environment with a single beam echo-sounder. This view presents two instants of the mission, before and after performing a loop. The robot trajectory is projected in blue on the sea-floor.

The problem of verify a loop is not trivial, even in two-dimensional contexts. A first proposition has already been the subject of [8], with a test based on the Newton operator [9]. However, this test $\mathcal{N}$ is not always able to conclude on obvious existence cases, as it stands on a Jacobian matrix that is sometimes not invertible. Our contribution is to propose a new test $\mathcal{T}$ relying on the topological degree theory [10, 11] that outperforms the previous method, thus increasing the number of proofs of loop closures on robot trajectories.

This paper is organized as follows. Section 2 details how loops can be detected thanks to proprioceptive measurements, especially in a bounded-error context. It is shown that proving the existence of a loop amounts to checking that an uncertain function vanishes at some point, which can be verified thanks to the topological degree theory presented in Section 3. This theoretical part applied on our loop problem is implemented under a new dedicated existence test provided in Section 4. The same tool is extended in Section 5 for uniqueness verification purposes in order to prove that a given detection set encloses a unique solution for a loop. The proposed algorithms are then applied on an actual experiment described in Section 6, before the conclusion of the paper. A discussion about the optimality of the method is appended to this document.

2. Proprioceptive loop detections

This section details how loops can be detected thanks to proprioceptive measurements only. We recall that *proprioceptive* measurements shall mean values about robot's states sensed by the robot itself, for instance: velocity, inertial values, heading, *etc.* A definition of a loop set is provided, before details about guaranteed tools that will be used then for loop detections in a bounded-error context.

2.1. Formalization

In [8], a loop is defined by a t-pair (t_1, t_2) such that $\mathbf{p}(t_1) = \mathbf{p}(t_2)$, $t_1 \neq t_2$, where $\mathbf{p}(t)$ is the two-dimensional position of the robot at t. The loop detection consists in computing the set $\mathbb{T}^*$ of all loops:

$$\mathbb{T}^* = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \mathbf{p}(t_1) = \mathbf{p}(t_2), t_1 < t_2\right\}, \quad (1)$$

with t_0, t_f being respectively the start and end times of the trajectory. Graphically, we represent the *loop set* $\mathbb{T}^*$ as a set of points in the t-plane. An example of $\mathbb{T}^* = \{(t_a, t_b), (t_c, t_f), (t_d, t_e)\}$ is provided in Figure 4.

We consider a mobile robot moving on a horizontal plane. Its trajectory is made of several 2D positions defined by

$$\mathbf{p}(t) = \int_{t_0}^{t} \mathbf{v}(\tau) d\tau + \mathbf{p}_0, \quad (2)$$

where $\mathbf{v}(t) \in \mathbb{R}^2$ is the velocity vector of the robot at time $t \in [t_0, t_f]$ expressed in the environment reference frame. $\mathbf{v}(t)$ is a *proprioceptive* information that can be easily sensed by the robot at any time. Then, the loop set $\mathbb{T}^*$ is

$$\mathbb{T}^* = \left\{(t_1, t_2) \in [t_0, t_f]^2 \mid \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau = \mathbf{0}, t_1 < t_2\right\}, \quad (3)$$

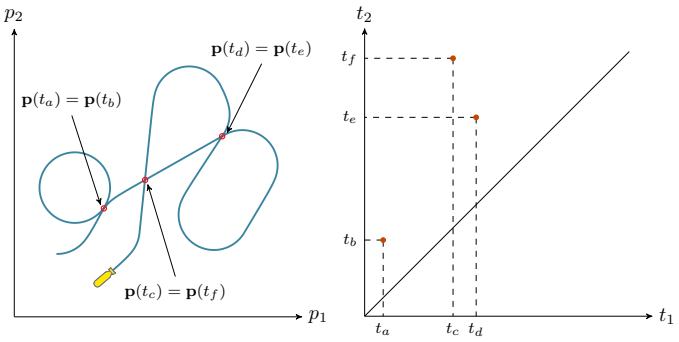


Figure 4: A robot performing three loops: its own trajectory has been crossed three times. A temporal representation provided by the t-plane (right-hand side) is used to picture the loops by t-pairs (t_a, t_b), (t_c, t_f), (t_d, t_e). The diagonal line corresponds to the *no-delay* line for which $t_1 = t_2$.

which means that for any $(t_1, t_2) \in \mathbb{T}^*$, robot's move from t_1 vanishes at t_2. Therefore, any loop can be detected based on these velocity measurements.

In practice, trajectories are estimated by noisy measurements which leads to spatial uncertainties. Hence, $\mathbb{T}^*$ cannot be computed exactly from Eq. (3). In what follows, we assume that the measurements are performed with a known bounded error [12], *i.e.* a box $[\mathbf{v}](t)$ contains the actual $\mathbf{v}^*(t)$ for each $t \in [t_0, t_f]$. This set-membership approach will stand on interval analysis, a mathematical field that appeared during the last decades [13] and is particularly suitable for verified computing. This tool is briefly presented hereinafter.

2.2. Tools for guaranteed computations

This section first introduces basic notions of interval analysis [9, 14] before focusing on tubes that will be used to handle proprioceptive measurements and their uncertainties over time.

2.2.1. Interval analysis

An interval $[x] = [x^-, x^+] = \{x \in \mathbb{R} \mid x^- \leqslant x \leqslant x^+\}$ is a closed and connected subset of $\mathbb{R}$ delimited by a lower bound x^- and an upper one x^+. A Cartesian product of n intervals defines a *box* – also called *interval-vector* – belonging to the set $\mathbb{IR}^n$. In this paper, intervals are written into brackets and vectors and boxes are represented in bold: $[\mathbf{x}]$. The actual but unknown value, enclosed within a box, is denoted by a star: $\mathbf{x}^*$.

Interval analysis is based on the extension of all classical real arithmetic operators $+$, $-$, $\times$ and $\div$. For instance:

$$[x] + [y] = [x^- + y^-, x^+ + y^+],$$
$$[x] - [y] = [x^- - y^+, x^+ - y^-].$$

This extension also includes the adaptation of elementary functions such as cos, exp, tan. The output is the smallest interval containing all the images of all defined inputs through the function.

2.2.2. Tubes

Classical intervals of reals can be extended to trajectories by means of *tubes*. A tube [15, 16] $[\mathbf{x}](\cdot) : \mathbb{R} \to \mathbb{IR}^n$ is an envelope enclosing an uncertain trajectory denoted by $\mathbf{x}^*(\cdot) : \mathbb{R} \to \mathbb{R}^n$. This enclosure can be defined as an interval of two functions $\mathbf{x}^-(\cdot)$ and $\mathbf{x}^+(\cdot)$ such that $\forall t \in [t_0, t_f], \mathbf{x}^-(t) \leqslant \mathbf{x}^+(t)$. Figure 5 gives an illustration of a tube enclosing a trajectory $x^*(\cdot) : \mathbb{R} \to \mathbb{R}$. In practice, a tube can be implemented as a set of boxes representing temporal slices. The enclosure of the tube is then piecewise constant, which allows classical operations in a simple and reliable way. An example of this implementation is given in [17].

As for intervals, tubes can be handled with the extension of classical real arithmetic operators (such as addition, usual functions, *etc.*). This can be done using interval arithmetic applied on each t of the definition domain $[t_0, t_1]$.

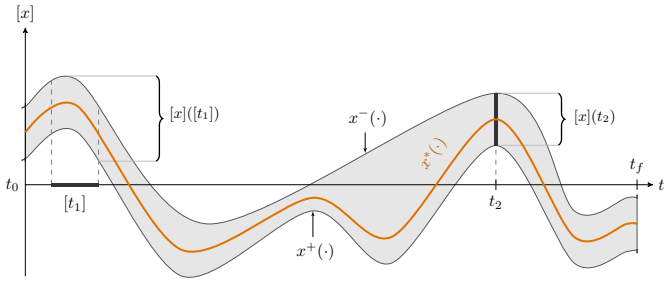


Figure 5: A tube $[x](\cdot)$ defined on $[t_0, t_f]$ as an interval of two functions $[x^-(\cdot), x^+(\cdot)]$ and enclosing an unknown trajectory $x^*(\cdot)$. The thinner the tube, the better the approximation of $x^*(\cdot)$.

The integral of a tube is defined from t_1 to t_2 as the smallest box containing all feasible integrals:

$$\int_{t_1}^{t_2} [\mathbf{x}](\tau)d\tau = \left\{ \int_{t_1}^{t_2} \mathbf{x}(\tau)d\tau \mid \mathbf{x}(\cdot) \in [\mathbf{x}](\cdot) \right\}. \quad (4)$$

From the monotonicity of the integral operator, we can deduce:

$$\int_{t_1}^{t_2} [\mathbf{x}](\tau)d\tau = \left[\int_{t_1}^{t_2} \mathbf{x}^-(\tau)d\tau, \int_{t_1}^{t_2} \mathbf{x}^+(\tau)d\tau \right]. \quad (5)$$

The lower bound of this box is illustrated by Figure 6. The integral can also be computed between bounded bounds $[t_1]$, $[t_2]$ by

$$\int_{[t_1]}^{[t_2]} [\mathbf{x}](\tau)d\tau = \begin{bmatrix} \text{lb}\left(\mathbf{y}^-([t_2]) - \mathbf{y}^-([t_1])\right), \\ \text{ub}\left(\mathbf{y}^+([t_2]) - \mathbf{y}^+([t_1])\right) \end{bmatrix}, \quad (6)$$

where $[\mathbf{y}](t) = \int_{t_0}^{t} [\mathbf{x}](\tau)d\tau$ is the interval primitive of $[\mathbf{x}](\cdot)$ and $\mathbf{y}^-(\cdot)$, $\mathbf{y}^+(\cdot)$ are the corresponding bounds. The proof is provided in [8, Sec. 3.3].

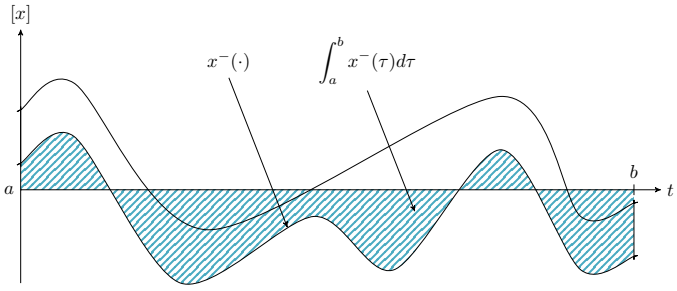


Figure 6: Lower bound of the integral of a tube. Hatched part depicts the lower bound of $\int_a^b [x](\tau)d\tau$.

A tube is generally used to describe uncertain trajectories evolving with time and defined by differential equations [18, 19, 20]. This is useful in mobile robotics where robots are expressed as uncertain dynamical systems submitted to temporal constraints such as trajectory evaluations or delays.

2.3. Loop detections in a bounded-error context

It has been shown in Section 2.1 that a loop can be detected based on velocity measurements. In practice, trajectories are estimated by measurements corrupted by noise, leading to spatial uncertainties. Hence, from Eq. (3), the set of t-pairs cannot be computed exactly. In a set-membership context [21, 22], measurement errors are bounded. In what follows, we assume that the actual values of the velocity $\mathbf{v}^*(\cdot)$ are unknown, but guaranteed to lie in the known tube $[\mathbf{v}](\cdot)$. The loop detection problem then amounts to computing the set $\mathbb{T}$ containing all feasible loops according to the given uncertainties:

$$\mathbb{T} = \left\{ (t_1, t_2) \mid \exists \mathbf{v}(\cdot) \in [\mathbf{v}](\cdot), \int_{t_1}^{t_2} \mathbf{v}(\tau)d\tau = \mathbf{0} \right\}, \quad (7)$$

or equivalently:

$$\mathbb{T} = \{(t_1, t_2) \mid \mathbf{0} \in [\mathbf{f}](t_1, t_2)\}, \quad (8)$$

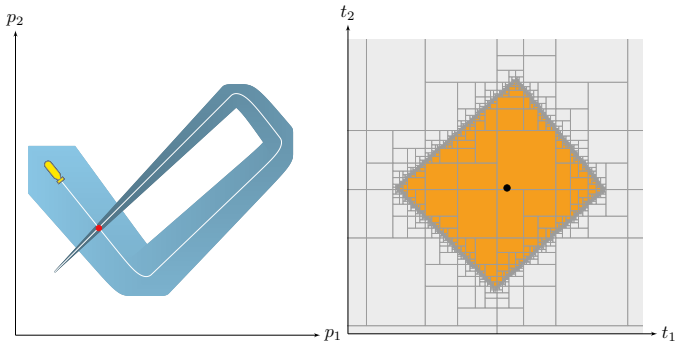
with $[\mathbf{f}] : \mathbb{IR}^2 \to \mathbb{IR}^2$ an inter-temporal inclusion function defined by

$$[\mathbf{f}]([t_1], [t_2]) = \int_{[t_1]}^{[t_2]} [\mathbf{v}](\tau)d\tau. \quad (9)$$

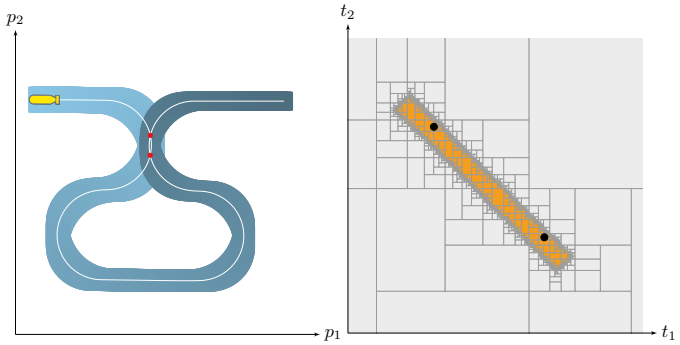
Hence, $\mathbb{T}$ is a reliable enclosure of $\mathbb{T}^*$ so that for each t-pair in $\mathbb{T}$, there exist values in the set of measurements that lead to the detection of a feasible loop. Therefore, the following relation is guaranteed:

$$\mathbb{T}^* \subseteq \mathbb{T} \subseteq [t_0, t_f]^2. \quad (10)$$

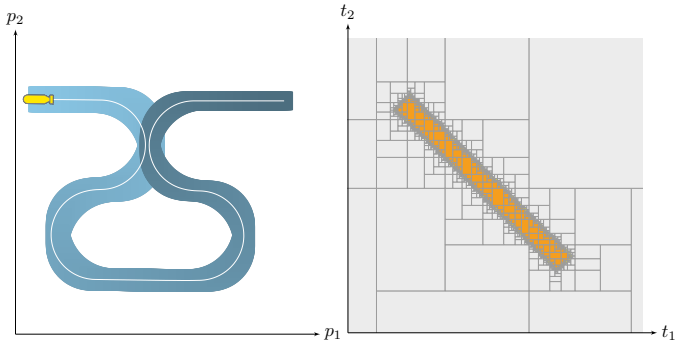
Figure 7 illustrates numerical approximations of $\mathbb{T}$ with a SIVIA algorithm [23, 8] over several examples. As it can be seen, the detection of a potential loop is not a proof of its existence. For instance, Figures 7b–7c are two identical cases regarding the uncertainties: the detection $\mathbb{T}$ pictured in the t-plane is the same while the actual trajectory may let appear one loop, two loops, or none.



(a) Loop detection over an undeniable looped trajectory.



(b) Loop detection over a doubtful looped trajectory. In this case the actual trajectory is made of two loops approximated within the same detection.



(c) Loop detection over a doubtful looped trajectory. In this case the actual trajectory never crosses itself despite a loop detection.

Figure 7: Guaranteed loop detections of a mobile robot. Several evolutions are drawn on the left hand side: the true trajectory is plotted in white while the computed envelope of all feasible trajectories is represented in blue, depicting some localization uncertainties due to measurement errors. Parts of the corresponding t-planes are pictured on the right hand side with the loop detection set $\mathbb{T}$ approximated by a set of boxes $[\mathbf{t}]_j$. This reliable approximation is obtained with a SIVIA algorithm. When an actual loop $\mathbf{t}^*$ exists – pictured by a black dot – it is surely enclosed by this set of boxes.

Note that depending on robot's trajectory, the numerical approximation of $\mathbb{T}$ may consist of several connected components denoted $\mathbb{T}_i$, see Figure 8.

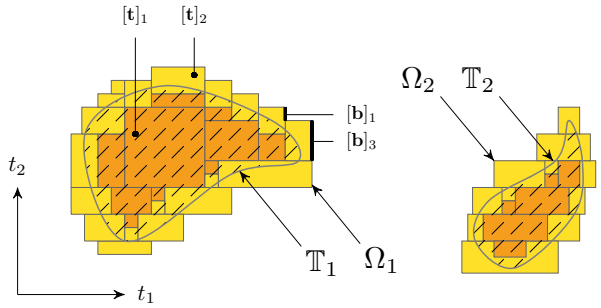


Figure 8: Approximation of a set $\mathbb{T} = \mathbb{T}_1 \cup \mathbb{T}_2$ with sets of non-overlapping boxes. In this paper, only the outer approximations Ω_i (unions of connected boxes called *subpavings*) will be assessed.

The only way to prove the existence of at least one loop in a given subset $\mathbb{T}_i$ is to verify that $\forall \mathbf{f} \in [\mathbf{f}], \exists (t_1, t_2) \in \mathbb{T}_i$ such that $\mathbf{f}(t_1, t_2) = \mathbf{0}$, which is equivalent to verifying a zero of an unknown function[1] $\mathbf{f}^* \in [\mathbf{f}]$ on $\mathbb{T}_i$. This can be shown using the Newton test $\mathcal{N}$ from [9]. Our contribution is to propose a new test $\mathcal{T}$ based on the topological degree that outperforms the previous method in most cases of ambiguous trajectories, *i.e.* non-robust zeros. This will be presented in Sections 3 and 4.

3. Topological degree for zeros verification

In what follows, we assume that an inclusion function $[\mathbf{f}] : \mathbb{IR}^n \to \mathbb{IR}^n$ of the unknown continuous function $\mathbf{f}^* : \mathbb{R}^n \to \mathbb{R}^n$ is given, possibly in the form of an algorithm for computing $[\mathbf{f}]([\mathbf{t}])$.

We want to isolate and verify (prove the existence of) zeros of $\mathbf{f}^*$. It immediately follows from the definition that if $\mathbf{0} \notin [\mathbf{f}]([\mathbf{t}])$ for some box $[\mathbf{t}]$, then $\mathbf{f}^*$ has no zero on $[\mathbf{t}]$. It is, however, harder to verify the *existence* of zero inside a region. If $\mathbf{0} \in [\mathbf{f}]([\mathbf{t}])$, we cannot disprove $\mathbf{f}^*(\mathbf{t}) = \mathbf{0}$ for some $\mathbf{t}$, but it is also not obvious how to prove the existence of such $\mathbf{t}$.

A powerful tool for verifying zeros is the topological degree, denoted by $\deg(\mathbf{f}^*, \Omega)$. It is a unique integer assigned to $\mathbf{f}^*$ and a compact set[2] $\Omega \subset \mathbb{R}^n$ such that $\mathbf{f}^*(\mathbf{t}) \neq \mathbf{0}$ for all $\mathbf{t} \in \partial\Omega$. In this definition, $\partial\Omega$ represents the boundary of the set Ω. The topological degree satisfies certain properties,

[1] The unknown function $\mathbf{f}^* : \mathbb{R}^2 \to \mathbb{R}^2$, defined as $\mathbf{f}^* = \int_{t_1}^{t_2} \mathbf{v}^*(\tau) d\tau$, cannot be evaluated as we do not know the actual velocity $\mathbf{v}^*(\cdot)$ of the robot.

[2] In some references such as [24], Ω is assumed to be open and bounded, which corresponds to considering the *interior* of our Ω. The requirement $\mathbf{f}^*(\mathbf{t}) \neq \mathbf{0}, \forall \mathbf{t} \in \partial\Omega$ is unchanged.

see [24, 10, 25] for detailed expositions. For our purposes, the most important property is that

$$\deg(\mathbf{f}^*, \Omega) \neq 0 \implies \exists \mathbf{t} \in \Omega \mid \mathbf{f}^*(\mathbf{t}) = \mathbf{0}. \quad (11)$$

Recent advances in computational topology generated many algorithms for computing the topological degree. Besides, it can be computed in case where only an inclusion function $[\mathbf{f}]$ of $\mathbf{f}^*$ is given. It was argued in [26, Sec. 9] that the degree test is in many cases more powerful than more classical verification tools including interval Newton, Miranda's or Borsuk's tests (see [27, 28, 29] for definitions of those tests). Our application for detecting robot loops deals with the two-dimensional case, $\Omega \subset \mathbb{R}^2$, for the reason that loops are defined by couples of times. Then the degree has a particularly nice geometric interpretation: it is the *winding number* of the curve $\partial\Omega \overset{\mathbf{f}^*}{\mapsto} \mathbb{R}^2 \setminus \{\mathbf{0}\}$ around $\mathbf{0}$, see Figure 9. If $[\mathbf{f}]$ is given, then the winding number can be computed by a number of elementary methods, the algorithm of [11] being one of them.

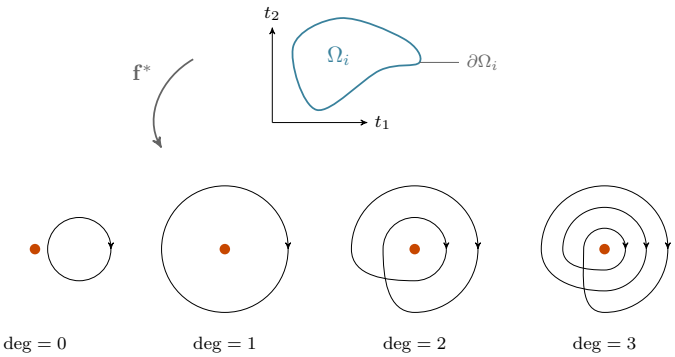


Figure 9: Computation of the degree of $\mathbf{f}^*$ on Ω_i. The illustration shows several positive degree cases.

Consider a given subdomain $\mathbb{T} \subset \mathbb{R}^n$ in which we want to find zeros of $\mathbf{f}^*$. For computational purposes, an outer approximation of $\mathbb{T}$ is performed by dividing the space into a set of non-overlapping boxes denoted $[\mathbf{t}]_j$. An algorithm relying on set inversion such as SIVIA [23] can be used to this end. Figure 8 depicts such reliable approximation. The outer set has the properties required for Ω. Consequently, the set Ω we consider will always be a finite union of boxes.

The following statement is a reformulation of [11, Theorem 2.9] adapted to our notation.

Theorem 1. *Let Ω be a union of finitely many boxes in $\mathbb{IR}^n$:*

$$\Omega = \bigcup_{j=1}^{l} [\mathbf{t}]_j, \quad (12)$$

and assume that its boundary $\partial\Omega$ is a union of finitely many boxes

$$\partial\Omega = \bigcup_{k=1}^{p} [\mathbf{b}]_k \ [3]. \quad (13)$$

If $\mathbf{0} \notin [\mathbf{f}]([\mathbf{b}]_k)$ for all $k = 1, \ldots, p$, then the degree $\deg(\mathbf{f}^*, \Omega)$ is uniquely determined and its computation can be done from the evaluations of $[\mathbf{f}]([\mathbf{b}]_k)$.

Under the assumptions of the theorem, it immediately follows that $\deg(\mathbf{g}, \Omega) = \deg(\mathbf{f}^*, \Omega)$ for any $\mathbf{g} \in [\mathbf{f}]$, because $[\mathbf{f}]$ is also an inclusion function for $\mathbf{g}$ in such case.

Let $\Omega_1, \ldots, \Omega_l$ be connected components of the union of such boxes $[\mathbf{t}]_j$ with potential zeros. On each Ω_i, if its boundary is covered by boxes $[\mathbf{b}]_k$ such that $\mathbf{0} \notin [\mathbf{f}]([\mathbf{b}]_k)$ for each k, we can compute $\deg(\mathbf{f}^*, \Omega_i)$. Whenever this degree is nonzero, we verified the existence of at least one $\mathbf{t} \in \Omega_i$ such that $\mathbf{f}^*(\mathbf{t}) = \mathbf{0}$. We emphasize that the function $\mathbf{f}^*$ was unknown and we only worked with its inclusion function $[\mathbf{f}]$.

In the above paragraph, we never used derivatives of $\mathbf{f}^*$. Using additional information on derivatives, we can also count the number of solutions. Namely, if Ω is connected and $\deg(\mathbf{f}^*, \Omega) = \ell$ and we further know that the Jacobian matrix $\mathbf{J}_{\mathbf{f}^*}$ is nonsingular everywhere on Ω, then $\mathbf{f}^*$ has *exactly* $|\ell|$ zeros in Ω. This immediately follows from the definition of the degree given, for example, in [30, p. 27]. In particular, if the degree is ± 1, then non-singularity immediately implies that there is a unique zero of $\mathbf{f}^*$ in Ω. More details about this remark are given in Section 5.

4. Loop existence test

The topological degree theory will be used for proving the existence of robot loops. This section provides the proposed *existence test* with an explicit algorithm.

4.1. From topological degree to loops proofs

The inclusion function $[\mathbf{f}]$ assumed in Section 3 is given by Eq. (9), computable with Eq. (6). A SIVIA algorithm relying on Eq. (9) provides an outer approximation Ω of the set $\mathbb{T}$ resulting in several subpavings denoted by Ω_i. Such algorithm provides guaranteed results given the inclusion function that can be built from datasets, see [23]. The following relation is then guaranteed:

$$\mathbb{T}^* \subseteq \mathbb{T} \subseteq \left(\bigcup_i \Omega_i \right) \subseteq [t_0, t_f]^2. \quad (14)$$

[3] We also consider degenerate boxes. In this case, $[\mathbf{b}]$'s are boxes in $\mathbb{IR}^n$ of topological dimension $n-1$.

Each of these subpavings Ω_i constitutes a potential loop detection: there exists at least one trajectory with a $\mathbf{v}(\cdot) \in [\mathbf{v}](\cdot)$ that looped for one t-pair belonging to Ω_i. However, the trajectory related to the actual but unknown $\mathbf{v}^*(\cdot)$ may have never looped in reality despite the detection, as pictured by Figure 7. As a consequence, proving a loop amounts to verifying a zero of $\mathbf{f}^* : \mathbf{t} \mapsto \int_{t_1}^{t_2} \mathbf{v}^*(\tau)d\tau$ in Ω_i using the known inclusion function given by Eq. (9). By using the topological degree in this context, the consequent of the implication given in Eq. (11) is a proof of a loop existence. The algorithm for numerical verification of $\deg(\mathbf{f}^*, \Omega_i) \neq 0$ is provided hereinafter.

4.2. Implementation

This section shows how to apply a simple version of the topological degree algorithm for the special case of a connected two-dimensional region Ω_i that consists of 2D boxes. The following algorithms are an adaptation of [11] for this special case.

Assume that $\Omega_i \subset \mathbb{R}^2$ is a union of finitely many boxes and the boundary $\partial\Omega_i$ is a topological circle[4]. Furthermore, let $\mathbf{a}_1 \ldots, \mathbf{a}_p$ be points in $\partial\Omega_i$ and $[\mathbf{b}]_1, \ldots, [\mathbf{b}]_p$ be edges covering the boundary $\partial\Omega_i$, such that $\partial[\mathbf{b}]_i = \{\mathbf{a}_{i+1}, \mathbf{a}_i\}$ for $i < p$ and $\partial[\mathbf{b}_p] = \{\mathbf{a}_1, \mathbf{a}_p\}$. We endow each $[\mathbf{b}]_i$ with an *orientation* such that $\mathbf{a}_{i+1}$ is an end-point of $[\mathbf{b}]_i$ and $\mathbf{a}_i$ is the starting-point of $[\mathbf{b}]_i$ for $i < p$ and, similarly, $\mathbf{a}_1$ is the end-point of $[\mathbf{b}]_p$ and $\mathbf{a}_p$ the starting-point of $[\mathbf{b}]_p$. We define the *oriented boundary* of $[\mathbf{b}]_i$ to be $\mathbf{a}_{i+1} - \mathbf{a}_i$ for $i < p$ and the oriented boundary of $[\mathbf{b}]_p$ to be $\mathbf{a}_1 - \mathbf{a}_p$, where we introduce *oriented vertices* $\pm\mathbf{a}_j$ as formal symbols. This structure of oriented edges and oriented vertices can easily be represented in a computer.

Further, assume that an interval function $[\mathbf{f}]$ is given such that $\mathbf{0} \notin [\mathbf{f}]([\mathbf{b}]_i)$ for all i. This means that either the first or the second coordinate of the box $[\mathbf{f}]([\mathbf{b}]_i)$ has a constant sign, $+$ or $-$. We assign to the oriented box $[\mathbf{b}]_i$ the pair (c_i, s_i) where $c_i \in \{1, 2\}$ and $s_i \in \{+, -\}$ in such a way that the c_i-th coordinate of $[\mathbf{f}]([\mathbf{b}]_i)$ has a constant sign s_i. For example, $(2, -)$ indicates that the second coordinate of $[\mathbf{f}]([\mathbf{b}]_i)$ is negative: in particular f_2^* is negative on $[\mathbf{b}]_i$. Such choice (c_i, s_i) is not necessarily unique, but any choice will give us a correct result at the end.

The degree $\deg(\mathbf{f}^*, \Omega_i)$ can be computed using the following algorithms. The existence test $\mathcal{T}$ is then a direct conclusion on the computed degree. One should note that, at this step, Algorithm 1 is not able to reject the feasibility of a loop. In case of a non-zero degree, it will prove a loop existence. Otherwise, the "$\varnothing$" output will reflect a non-conclusive test.

[4]Hence, we shall assume that the set Ω_i is strictly included in $[t_0, t_f]^2$ so that a closed boundary $\partial\Omega_i$ can be assessed.

Algorithm 1 existenceTest$\mathcal{T}$ (in : $\Omega_i, [\mathbf{f}]$ − out : true$|\varnothing$)

```
1: begin
2: [b]_1 ... [b]_p ← getContour (Ω_i)
3: if 2dTopoDegree ([b]_1 ... [b]_p, [f]) ≠ 0 then
4:    return true
5: else
6:    return ∅    // not able to conclude about existence
7: end if
8: end
```

Algorithm 2 2dTopoDegree (in : $[\mathbf{b}]_1 \ldots [\mathbf{b}]_p, [\mathbf{f}]$ − out : d)

```
1: begin
2: d ← 0
3: for i = 1 to p do
4:    (c_i, s_i) ← tagEdge ([b]_i, [f])
5: end for
6: c_0 ← c_p, s_0 ← s_p, c_{p+1} ← c_1, s_{p+1} ← s_1
7: for i = 1 to p do
8:    if (c_i, s_i) = (1, +) then
9:       if (c_{i+1}, s_{i+1}) = (2, +) then
10:         d ← d + 1
11:      end if
12:      if (c_{i−1}, s_{i−1}) = (2, +) then
13:         d ← d − 1
14:      end if
15:   end if
16: end for
17: return d
18: end
```

An illustration of Algorithm 2 is given in Figure 10. Here the algorithm returns zero, because the if-conditions are satisfied only for the edge $[\mathbf{b}]_1$ where d will change from 0 to -1, and then in edge $[\mathbf{b}]_4$ where d will be changed from -1 to 0.

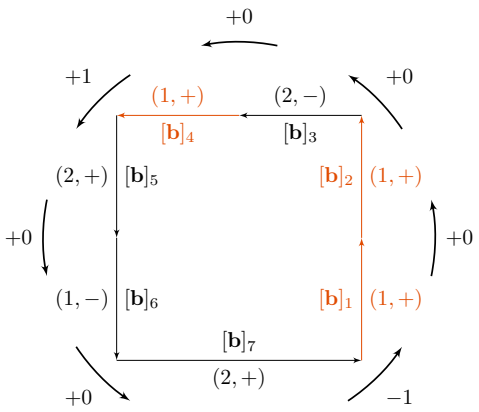


Figure 10: Illustration of the degree algorithm. The *selected* edges in this case are $[\mathbf{b}]_1, [\mathbf{b}]_2, [\mathbf{b}]_4$ but only $[\mathbf{b}]_1$ results in an addition by -1 and $[\mathbf{b}]_4$ in an addition of $+1$. The overall degree is $1 - 1 + 5 \times 0 = 0$ in this case.

If our representation of Ω_i comes from the previous SIVIA

Algorithm 3 tagEdge (in : $[\mathbf{b}], [\mathbf{f}] -$ out : (c, s))

begin
if $0 \notin [f_1]([\mathbf{b}])$ **then**
  if $[f_1]([\mathbf{b}]) \subset \mathbb{R}^+$, **return** $(1, +)$
  else, **return** $(1, -)$
else if $0 \notin [f_2]([\mathbf{b}])$ **then**
  if $[f_2]([\mathbf{b}]) \subset \mathbb{R}^+$, **return** $(2, +)$
  else, **return** $(2, -)$
else
  return $\varnothing$ *// note: this case should not happen*
end if
end

Algorithm 4 loopsNumber (in : $\Omega_i, [\mathbf{f}], [\mathbf{J_f}] -$ out : ℓ)

begin
$[\mathbf{t}]_1 \dots [\mathbf{t}]_j \leftarrow \text{getBoxes}(\Omega_i)$
for $k = 1$ to j **do**
  if $0 \in \det([\mathbf{J_f}]([\mathbf{t}]_k))$ **then**
    return $\varnothing$
  end if
end for
$[\mathbf{b}]_1 \dots [\mathbf{b}]_p \leftarrow \text{getContour}(\Omega_i)$
$\ell \leftarrow \text{2dTopoDegree}([\mathbf{b}]_1 \dots [\mathbf{b}]_p, [\mathbf{f}])$
return $|\ell|$
end

algorithm, we can assume that the `getContour` function (in Algorithm 1) is available and has linear time-complexity. A naive implementation of Algorithm 2 has quadratic complexity. Its input $[\mathbf{b}]_1, \dots [\mathbf{b}]_p$ can be ordered and oriented in $\sim p^2$ steps so that the end-point of $[\mathbf{b}]_j$ (resp. $[\mathbf{b}]_p$) coincides with the starting-point of $[\mathbf{b}]_{j+1}$ (resp. $[\mathbf{b}]_1$). The rest then amounts to finding the signs (c_j, s_j) in one pass over all j and adding 1 (resp. -1) to a global variable whenever $(c_j, s_j) = (1, +)$ and the next (resp. previous) sign is $(2, +)$. A better implementation in $\mathcal{O}(p)$ is possible if we can access additional information, such as the boundary orientation of $[\mathbf{b}]_j$ induced from $\partial\Omega_i$.

5. Reliable number of loops

Aside from proving the existence of a loop, it may be interesting to count the number of solutions. This can be done using additional information on the derivatives. To this end, the Jacobian matrix $\mathbf{J_{f^*}}$ of the unknown $\mathbf{f}^*$ has to be approximated by $[\mathbf{J_f}]$. From Leibniz's integral rule,

$$[\mathbf{J_f}]([\mathbf{t}]) = \begin{pmatrix} \frac{\partial [f_1]}{\partial [t_1]} & \frac{\partial [f_1]}{\partial [t_2]} \\ \frac{\partial [f_2]}{\partial [t_1]} & \frac{\partial [f_2]}{\partial [t_2]} \end{pmatrix} = \begin{pmatrix} -[v_1]([t_1]) & [v_1]([t_2]) \\ -[v_2]([t_1]) & [v_2]([t_2]) \end{pmatrix}, \tag{15}$$

where $[\mathbf{v}](\cdot)$ is the tube containing the unknown velocity $\mathbf{v}^*(\cdot)$ of the robot.

If Ω_i is a compact set as defined in Section 3 and if the Jacobian matrix $\mathbf{J_{f^*}}$ is nonsingular everywhere on Ω_i, then the absolute value of the degree is the exact number of solutions for $\mathbf{f}^* = \mathbf{0}$ in Ω_i.

Proving the non-singularity of the Jacobian matrix amounts to verifying that its determinant is non-zero. Using the inclusion function from Eq. (15), this is equivalent to verifying $0 \notin \det([\mathbf{J_f}])$.

Algorithm 4 provided hereinafter returns the exact number of loops in a set Ω_i when the zeros are robust enough. Otherwise, nothing can be concluded regarding the uncertainties of the information.

Remark 2. *The algorithm used to compute the set Ω_i may provide wide boxes $[\mathbf{t}]_k$ that will result in an over-approximation of the $[\mathbf{J_f}]([\mathbf{t}]_k)$. A bisection of the $[\mathbf{t}]_k$ may be applied when $0 \in \det([\mathbf{J_f}]([\mathbf{t}]_k))$ in order to deal with smaller boxes, thus reducing the pessimism of the Jacobian evaluation and increasing the chances to disprove $0 \in \det([\mathbf{J_f}]([\mathbf{t}]_k))$. If the determinant approximation still contains 0 beyond a given precision, then the algorithm should stop being not able to conclude.*

6. Application on real datasets

The efficiency of the proposed test is demonstrated over two experiments involving actual underwater robots. The underwater case is challenging as robots do not benefit from GNSS fixes except at the very beginning of the mission. Hence, dead-reckoning methods usually apply for state estimation, leading to strong cumulative errors. Loops will be proven in this context.

6.1. Absolute velocities

Underwater robots are usually equipped with an Inertial Measurement Unit (IMU) providing the Euler angles (ψ, θ, φ) depicting the orientation of the robot. In addition, a Doppler Velocity Log (DVL) will track the vehicle's speed $\mathbf{v}_r \in \mathbb{R}^3$ over the seabed by acoustic means, providing values in robot's own coordinate system. The absolute speed vector $\mathbf{v} \in \mathbb{R}^3$, expressed in the environment reference frame, is then obtained by

$$\mathbf{v} = \mathbf{R}(\psi, \theta, \varphi) \cdot \mathbf{v}_r, \tag{16}$$

where $\mathbf{R}(\psi, \theta, \varphi)$ is a classical Euler matrix. For more details about state equations for underwater robots, one can refer to [31].

6.2. From sensors to reliable results

6.2.1. Obtaining bounded measurements at time t

In practice, a measurement error is often modeled by a Gaussian distribution which has an infinite support. Therefore, setting bounds around this measurement already constitutes a theoretical risk of loosing the actual value. A choice has to be made at this step, considering such risk. After that, however, any algorithm standing on interval methods is ensured to not increase this risk.

Data-sheets usually give sensor specifications such as the standard deviation σ. Hence, a measurement v_1 is assumed to belong to an interval $[v_1]$ centered on v_1 and inflated according to the sensor uncertainties. For instance, $[v_1] = [v_1 - 2\sigma, v_1 + 2\sigma]$ will provide a 95% confidence rate over the actual and unknown value v_1^*, considering the Gaussian distribution.

6.2.2. From measurements to tubes

Common sensors provide us only with a set of measurement vectors sampled over finitely many time values, while our algorithm deals with continuous interval functions. Our choice is to build a tube from this data by computing a piecewise linear interpolation $\mathbf{v}^{PL}(\cdot)$ between the measurements. We then create a tube $[\mathbf{v}](\cdot)$ such that[5]

$$[\mathbf{v}](\cdot) = \mathbf{v}^{PL}(\cdot) + [-2\sigma, 2\sigma]^2. \tag{17}$$

Note that some sensors may provide real-time evaluations of σ, depending on the uncertainties of the environment[6]. In this case, $[\mathbf{v}](\cdot)$ can also be built with a reliable non-constant thickness.

Practically, the time-sampling is much finer than any sudden velocity change and it is realistic to assume that the error $\mathbf{v}^{PL}(\cdot) - \mathbf{v}^*(\cdot)$ is approximately normally distributed and centered at zero. An example of a tube $[v_1](\cdot)$ is provided in Figure 11.

The interval function $[\mathbf{f}]$ used for loop detection is then defined with Eq. (9) as the integral of $[\mathbf{v}](\cdot)$.

[5]In fact, in our implementation, we enclose $\mathbf{v}^{PL}(\cdot)$ by an even larger neighborhood. Our choice is to build the tube as a set of boxes representing slices. We first subdivide $[t_0, t_f]$ into a set of small sub-intervals $[t_k, t_{k+1}]$ corresponding to groups of several velocity measurements. We then define each slice as a box $[t_k, t_{k+1}] \times \left([-2\sigma, 2\sigma]^2 + \cup_{t=t_k}^{t_{k+1}} \mathbf{v}^{PL}(t)\right)$.

[6]With DVL for instance, the velocity estimations are highly related to the altitude of the sensor over the seabed and the assumed knowledge of the water column, through which acoustic signals are propagated.

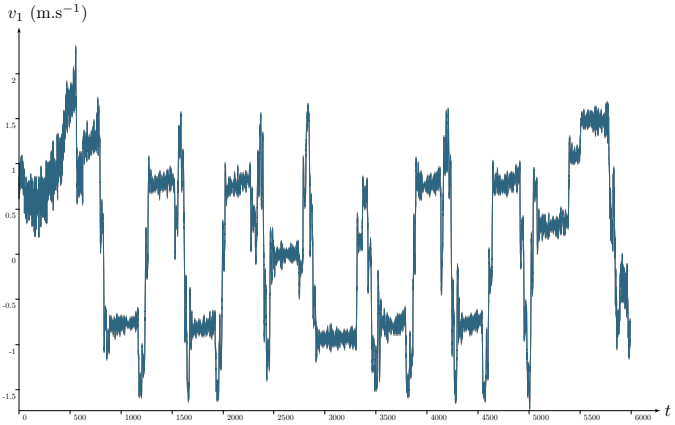


Figure 11: Tube $[v_1](\cdot)$ enclosing the *Redermor*'s east velocity. Despite appearances, the signal is not noised: the temporal domain of 1H40 is compressed to fit the tube in the figure.

Our method for loop detection is reliable under the assumption $\mathbf{f}^*([\mathbf{t}]) \subseteq [\mathbf{f}]([\mathbf{t}])$. This inclusion immediately follows from the assumption $\mathbf{v}^*(\cdot) \subseteq [\mathbf{v}](\cdot)$ but in fact, the former inclusion is much more robust with respect to random velocity errors than the latter.[7] A quantitative analysis of error probabilities is a work in progress.

6.3. *The* Redermor *mission*

This first application involves an Autonomous Underwater Vehicle (AUV) named *Redermor*, see Figure 12. This test case has already been the subject of [8, Sec. 6], in which the existence of 14 loops had been proved by using the test $\mathcal{N}$ relying on the Newton operator. Our goal is to compare these results with the topological degree test $\mathcal{T}$ we propose in this paper.

A two hours experimental mission has been done in the Douarnenez bay in Brittany (France). A top view of the area covered by the robot is pictured in Figure 13. *Redermor* performed 28 loops, 20m deep. The set-membership approach provides the enclosure of $\mathbf{v}^*(\cdot)$, see Figure 11, and then the approximation of $\mathbb{T}$ pictured in the t-plane of Figure 14. A total of 25 complete loop-detection sets have been computed on this test-case, the other solutions being partial. By *complete detections* we mean loop detection sets Ω_i strictly included in the t-plane. Further comments on this application will only stand on these detections and the related actual loops.

In both Figures 13 and 14, the result of the degree test is displayed in green when it proves the existence of a

[7]The real displacement $\int_{t_a}^{t_b} \mathbf{v}^*(\tau)\, d\tau$ could lie outside $[\mathbf{f}](t_a, t_b)$ only if the velocity errors would *cumulate in one direction*. More precisely, the projection of $\mathbf{v}^{PL}(\cdot) - \mathbf{v}^*(\cdot)$ into one particular direction would have to be at least 2σ *in average*, over the whole time interval $[t_a, t_b]$. Under fairly general assumptions on the distribution of the velocity errors, such probability decreases exponentially with $(t_b - t_a)$.

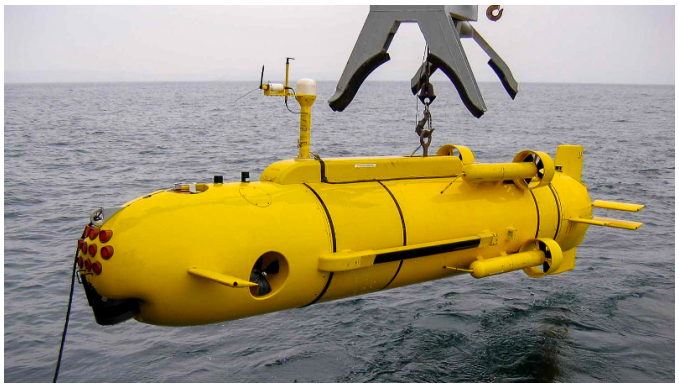


Figure 12: The *Redermor* autonomous underwater robot before a sea trial. This experiment has been done with the kind help of *DGA Techniques Navales Brest* (French Ministry of Defense).

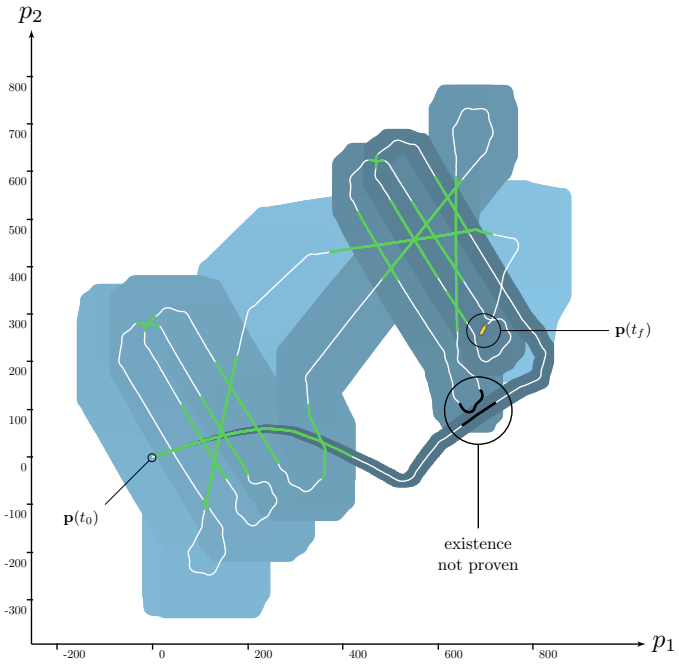


Figure 13: 2D trace of *Redermor* AUV. The projected tube $[\mathbf{p}](\cdot)$ (*i.e.* the bounded estimated positions) is drawn in blue, depicting an increasing localization uncertainty. The truth is plotted by the white line while green and black lines are the projections of the results given by the topological degree test $\mathcal{T}$.

loop and in black when nothing can be concluded. This latter case means the robot's uncertainties are too large to demonstrate that a loop has been performed or not. In this example, there is only one situation for which nothing can be concluded. If we have a look at Figure 13, we can see this inconclusive case, black painted above robot's trajectory. Figure 15 provides another view of it. Looking at the reliable envelope of feasible positions pictured in gray, it could have been a loop. We know it is not the case in reality: actual trajectories are not crossing. Here, the test does not reject the feasibility of a loop, it is simply not able to conclude.

We define the actual number of loops λ^* over a mission

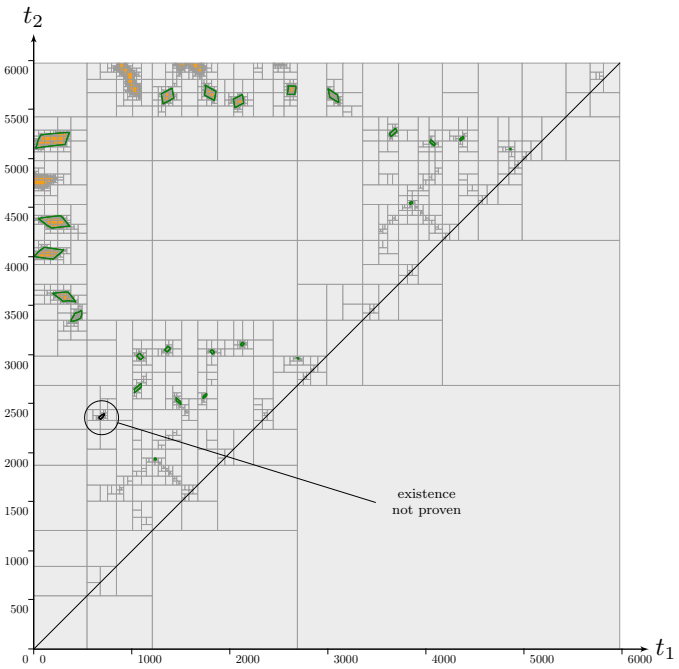


Figure 14: t-plane corresponding to the *Redermor* mission and computed with a SIVIA algorithm. There exist four partial detections Ω_i on t-plane's edges that will not be considered here since the $\partial\Omega_i$ are not totally defined. They enclose feasible loops (t_a, t_b) performed at the very beginning of the mission ($t_a \simeq t_0$) or at the end ($t_b \simeq t_f$).

by:

$$\lambda^* = \#\big\{\mathbf{t} \mid \mathbf{f}^*(\mathbf{t}) = \mathbf{0}, t_1 < t_2\big\}. \tag{18}$$

Now, considering uncertainties from the sensors, the theoretical number of provable loops is given by:

$$\lambda = \#\big\{\mathbb{T}_i \mid \forall \mathbf{f} \in [\mathbf{f}], \exists \mathbf{t} \in \mathbb{T}_i \mid \mathbf{f}(\mathbf{t}) = \mathbf{0}\big\}. \tag{19}$$

This application gives a comparison between the tests $\mathcal{T}$ and $\mathcal{N}$. Corresponding computations provide the following results:

$$\lambda_{\mathcal{N}} = 14 \qquad \lambda_{\mathcal{T}} = 24 \qquad \lambda^* = 24$$

The white line in Figure 13 shows that the actual trajectory involves $\lambda^* = 24$ loops[8]. On this application, no other test than the topological degree would provide better results.

6.4. The Daurade *mission*

We provide a complementary example involving another AUV named *Daurade*, pictured in Figure 16. A similar mission has been performed without surfacing during 1h40. Figure 17 presents the corresponding trajectory together with its estimation and the test results. Figure 18 and 19 provide views of the t-plane.

[8]Without considering the four loops in the components Ω_i that intersect the boundary of $[t_0, t_f]^2$.

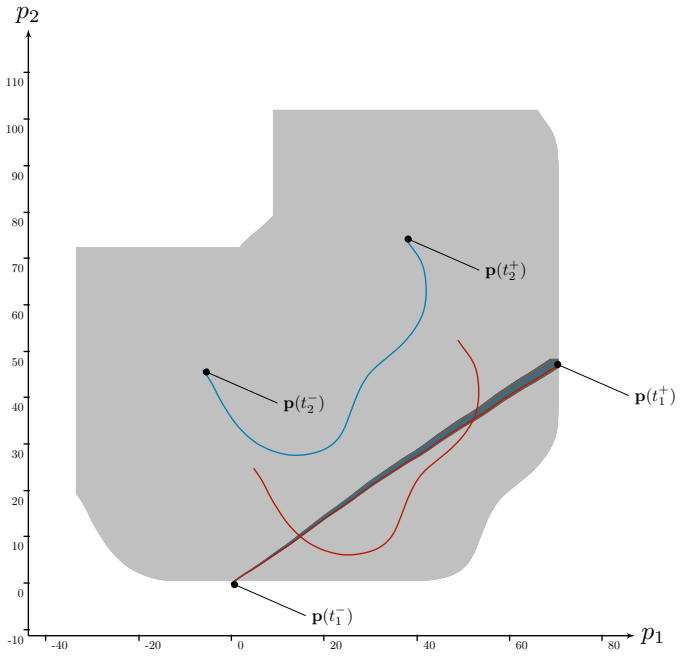


Figure 15: Independent projection of the non-conclusive case. Let us consider the loop-box $[t_1^-, t_1^+] \times [t_2^-, t_2^+]$ enclosing the corresponding Ω_i approximation. The actual trajectory over both $[t_1^-, t_1^+]$ and $[t_2^-, t_2^+]$ is plotted in blue. The bounded approximation is pictured in dark gray for the first part and light gray then. Note that we do not represent the amount of uncertainties gathered before t_1^-: $\mathbf{p}(t_1^-)$ is centered in $(0, 0)$ in this independent view. However, the amount of uncertainties over $[t_2^-, t_2^+]$ is such that other crossing trajectories would have been possible given the assumed uncertainties, see *e.g.* the red one. This shows the impossibility to both disprove this loop detection and conclude about a loop existence.

Figure 16: *Daurade* AUV managed by *DGA Techniques Navales Brest* and the *Service Hydrographique et Océanographique de la Marine* (SHOM), during an experiment dedicated to this work, in the Rade de Brest, October 2015.

For this test case, 116 subpavings Ω_i have been computed. The test $\mathcal{T}$ proved the existence of loops in 114 of them. The uniqueness was also verified for each proof. Computations have been performed in less than one second on a conventional computer, which also demonstrates the relevancy of our approach for real applications.

The actual trajectory involved $\lambda^* = 118$ loops[8] while we proved $\lambda_{\mathcal{T}} = 114$ of them. For two loop detection sets, the algorithm did not conclude due to strong uncertainties. One of these cases is highlighted in Figure 20. The conclusion is that in this *Daurade* experiment, no more loops would have been proved by other means than the topological degree.

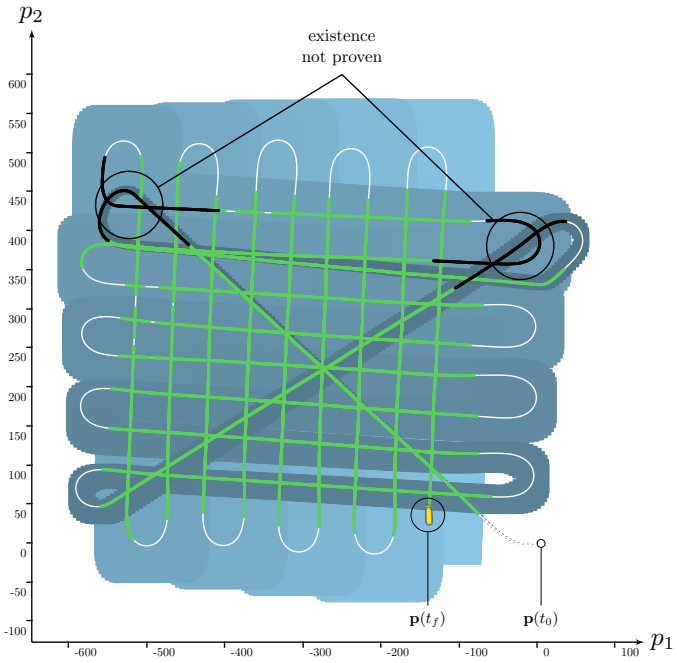


Figure 17: 2D trace of *Daurade* AUV. The topological test was not able to conclude for two loop detections involving a total of four actual loops. Figure 20 details one of these cases.

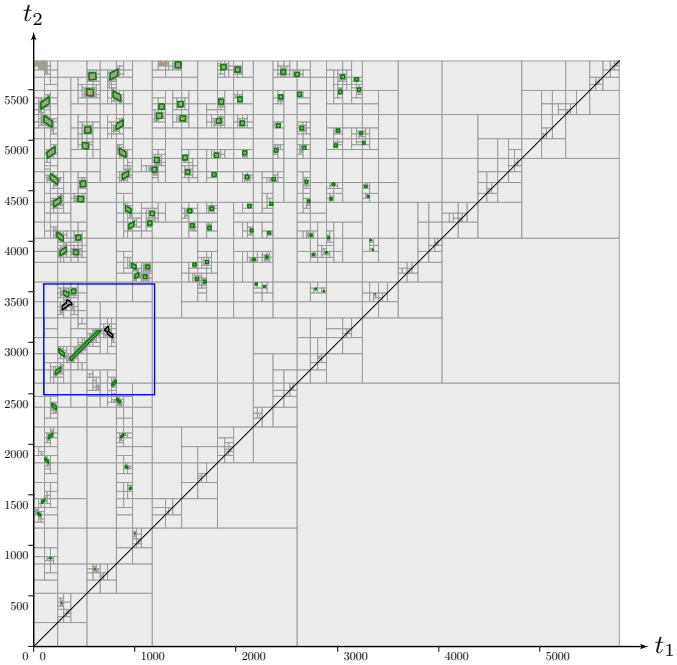


Figure 18: t-plane of the *Daurade* experiment. The blue box is detailed in Figure 19.

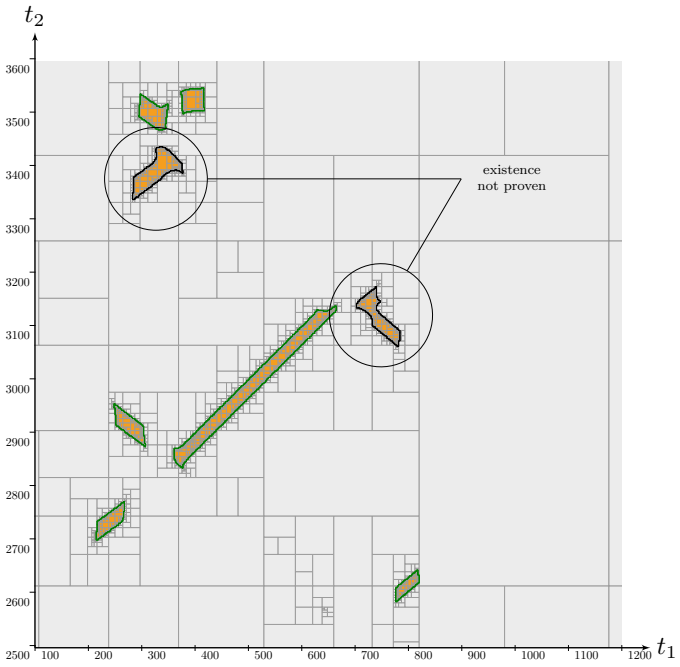


Figure 19: Zoom on t-plane of Figure 18, presenting eight clusters Ω_i corresponding to loop detection sets. Two of them, black painted, are non-conclusive cases with the topological degree test.

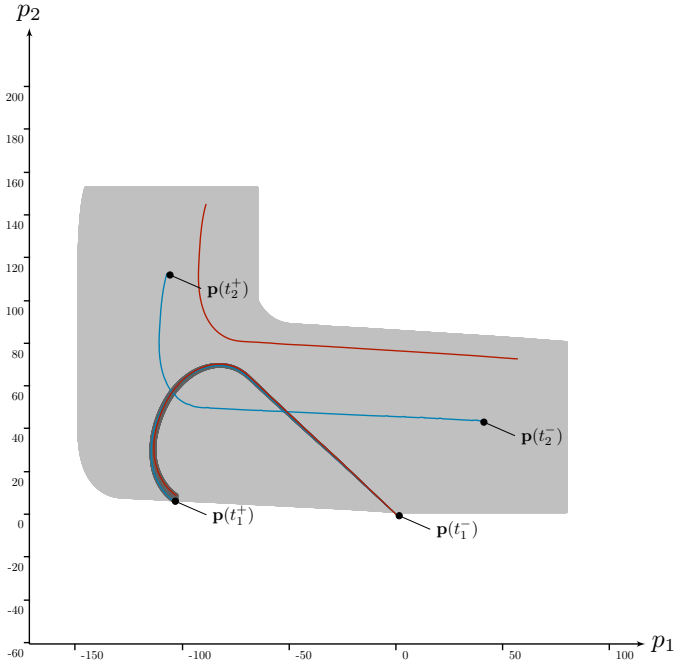


Figure 20: Independent projection of one of the two non-conclusive detection cases, as for the *Redermor* mission, see Figure 15. Contrary to the previous experiment, an actual loop plotted in blue has been performed, twice. However, the red trajectory reminds that a non-crossing case is still feasible.

7. Conclusion

This paper has presented a new method to prove the existence of loops in robot trajectories. The algorithm relies on interval analysis, allowing guaranteed computations of robot trajectories by considering sensor uncertainties in a reliable way. This set-membership approach stands on measurements' bounds which allow to take conclusions by always considering worst-case possibilities. This is well suited for proof purposes and, in our case, to prove that a robot crossed its own trajectory at some point. In this approach, conclusions can be taken considering proprioceptive measurements only and no scene observation. This is helpful to solve SLAM problems as it proves a previously-visited location to be recognized.

This topic has already been the subject of previous work but the offered *existence test*, relying on the Newton operator, did not give satisfactory results in some cases of undeniable looped trajectories. This was due to the use of Jacobian matrices not always invertible. Our contribution has been to propose a new test relying on the topological degree theory. The algorithm behaves better as it does not use the information of the derivatives. Besides the loop existence proof, the same tool can provide the exact number of reliable loops performed by the robot, better than the Newton test did. The efficiency of the new method has been demonstrated on actual experiments involving autonomous underwater robots performing several loops under the surface.

Supplementary materials are available on: `http://simon-rohou.fr/research/loopproof/`

8. Acknowledgments

The work of Simon Rohou has been funded by the French *Direction Générale de l'Armement* (DGA) during the UK-France PhD program. The one of Peter Franek has been supported by Austrian Science Fond, M 1980.

[1] R. Smith, M. Self, P. Cheeseman, Estimating Uncertain Spatial Relationships in Robotics, in: Autonomous Robot Vehicles, Springer, New York, NY, 1990, pp. 167–193. doi:10.1007/978-1-4613-8997-2_14.
URL https://link.springer.com/chapter/10.1007/978-1-4613-8997-2_14

[2] M. Bosse, P. Newman, J. Leonard, S. Teller, Simultaneous localization and map building in large-scale cyclic environments using the atlas framework, The International Journal of Robotics Research 23 (12) (2004) 1113–1139.

[3] Y. Latif, C. Cadena, J. Neira, Robust loop closing over time for pose graph slam, The International Journal of Robotics Research 32 (14) (2013) 1611–1626.

[4] A. Angeli, D. Filliat, S. Doncieux, J.-A. Meyer, Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words, IEEE Transactions on Robotics 24 (5) (2008) 1027–1037. doi:10.1109/TRO.2008.2004514.
URL http://ieeexplore.ieee.org/document/4633680/

[5] M. Cummins, P. Newman, FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance, The International Journal of Robotics Research 27 (6) (2008) 647–665. doi:10.1177/0278364908090961.
URL http://dx.doi.org/10.1177/0278364908090961

[6] C. Stachniss, D. Hahnel, W. Burgard, Exploration with active loop-closing for FastSLAM, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Vol. 2, 2004, pp. 1505–1510 vol.2. doi:10.1109/IROS.2004.1389609.

[7] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, J. D. Tardós, Mapping Large Loops with a Single Hand-Held Camera, in: Robotics: Science and Systems, Vol. 2, 2007.

[8] C. Aubry, R. Desmare, L. Jaulin, Loop detection of mobile robots using interval analysis, Automatica 49 (2) (2013) 463–470. doi:10.1016/j.automatica.2012.11.009.
URL http://linkinghub.elsevier.com/retrieve/pii/S0005109812005456

[9] R. Moore, Methods and Applications of Interval Analysis, Studies in Applied and Numerical Mathematics, Society for Industrial and Applied Mathematics, 1979.
URL https://books.google.fr/books?id=WYjD2-R2zMgC

[10] D. O'Regan, Y. J. Cho, Y. Q. Chen, Topological degree theory and applications, no. 10 in Series in mathematical analysis and applications, Chapman & Hall/CRC, Boca Raton, FL, 2006, oCLC: ocm64592216.

[11] P. Franek, S. Ratschan, Effective topological degree computation based on interval arithmetic, Mathematics of Computation 84 (293) (2014) 1265–1290. doi:10.1090/S0025-5718-2014-02877-9.
URL http://www.ams.org/mcom/2015-84-293/S0025-5718-2014-02877-9/

[12] D. Meizel, A. Preciado-Ruiz, E. Halbwachs, Estimation of mobile robot localization: geometric approaches, in: M. Milanese, J. Norton, H. Piet-Lahanier, E. Walter (Eds.), Bounding Approaches to System Identification, Plenum Press, New York, NY, 1996, pp. 463–489.

[13] R. Moore, Interval analysis, Prentice-Hall series in automatic computation, Prentice-Hall, 1966.
URL https://books.google.fr/books?id=csQ-AAAAIAAJ

[14] E. R. Hansen, Interval arithmetic in matrix computations - part I, SIAM Journal on Numerical Analysis: Series B 2 (2) (1965) 308–320.

[15] T. F. Filippova, A. B. Kurzhanski, K. Sugimoto, I. Vályi, Ellipsoidal State Estimation for Uncertain Dynamical Systems, in: M. Milanese, J. Norton, H. Piet-Lahanier, É. Walter (Eds.), Bounding Approaches to System Identification, Springer US, Boston, MA, 1996, pp. 213–238.
URL https://doi.org/10.1007/978-1-4757-9545-5_14

[16] F. Le Bars, J. Sliwka, L. Jaulin, O. Reynet, Set-membership state estimation with fleeting data, Automatica 48 (2) (2012) 381–387. doi:10.1016/j.automatica.2011.11.004.
URL http://linkinghub.elsevier.com/retrieve/pii/S0005109811005322

[17] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres, Guaranteed computation of robot trajectories, Robotics and Autonomous Systems 93 (2017) 76–84. doi:https://doi.org/10.1016/j.robot.2017.03.020.
URL http://www.sciencedirect.com/science/article/pii/S0921889016304006

[18] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres, Reliable non-linear state estimation involving time uncertainties, Automatica 93 (2018) 379–388. doi:https://doi.org/10.1016/j.automatica.2018.03.074.
URL https://www.sciencedirect.com/science/article/pii/S0005109818301699

[19] T. Raïssi, N. Ramdani, Y. Candau, Set membership state and parameter estimation for systems described by nonlinear differential equations, Automatica 40 (10) (2004) 1771–1777. doi:10.1016/j.automatica.2004.05.006.
URL http://www.sciencedirect.com/science/article/pii/S0005109804001529

[20] A. Goldsztejn, W. Hayes, P. Collins, Tinkerbell Is Chaotic, SIAM Journal on Applied Dynamical Systems 10 (4) (2011) 1480–1501. doi:10.1137/100819011.
URL http://epubs.siam.org/doi/10.1137/100819011

[21] V. Drevelle, P. Bonnifait, High integrity GNSS location zone characterization using interval analysis, in: ION GNSS 2009, Savannah, GA, United States, 2009, pp. 2178–2187.
URL https://hal.archives-ouvertes.fr/hal-00444819

[22] F. Abdallah, A. Gning, P. Bonnifait, Box particle filtering for nonlinear state estimation using interval analysis, Automatica 44 (3) (2008) 807–815. doi:10.1016/j.automatica.2007.07.024.
URL http://www.sciencedirect.com/science/article/pii/S0005109807003731

[23] L. Jaulin, É. Walter, Set inversion via interval analysis for nonlinear bounded-error estimation, Automatica 29 (4) (1993) 1053–1064. doi:10.1016/0005-1098(93)90106-4.
URL http://linkinghub.elsevier.com/retrieve/pii/0005109893901064

[24] I. Fonseca, W. Gangbo, Degree theory in analysis and applications, no. 2 in Oxford lecture series in mathematics and its applications, Clarendon Press ; Oxford University Press, Oxford : New York, 1995.

[25] M. Furi, M. Pera, M. Spadini, A Set of Axioms for the Degree of a Tangent Vector Field on Differentiable Manifolds, Fixed Point Theory and Applications 2010 (1) (2010) 845631. doi:10.1155/2010/845631.
URL http://www.fixedpointtheoryandapplications.com/content/2010/1/845631

[26] P. Franek, S. Ratschan, P. Zgliczynski, Quasi-decidability of a Fragment of the First-Order Theory of Real Numbers, Journal of Automated Reasoning 57 (2) (2016) 157–185. doi:10.1007/s10817-015-9351-3.
URL https://doi.org/10.1007/s10817-015-9351-3

[27] R. E. Moore, A Test for Existence of Solutions to Nonlinear Systems, SIAM Journal on Numerical Analysis 14 (4) (1977) 611–615. doi:10.1137/0714040.
URL http://epubs.siam.org/doi/10.1137/0714040

[28] R. E. Moore, J. B. Kioustelidis, A simple test for accuracy of approximate solutions to nonlinear (or linear) systems, SIAM Journal on Numerical Analysis 17 (4) (1980) 521–529.

[29] K. Borsuk, Drei Sätze über die n-dimensionale euklidische Sphäre, Fundamenta Mathematicae 20 (1) (1933) 177–190.
URL http://eudml.org/doc/212624

[30] J. W. Milnor, Topology from the differentiable viewpoint, rev. ed Edition, Princeton landmarks in mathematics, Princeton University Press, Princeton, N.J, 1997.

[31] T. I. Fossen, Guidance and control of ocean vehicles, Wiley, Chichester ; New York, 1994.

Appendix: Optimality of the degree test

In this complementary section, we extend the aforementioned practical demonstration by a theoretical discussion of the degree test and its strength.

First of all, in a situation where the interval Newton test $\mathcal{N}$ is strong enough to detect a (unique) solution of $\mathbf{f}^*(\mathbf{x}) = \mathbf{0}$ in a connected region Ω, then the Jacobian matrix $\mathbf{J}_{\mathbf{f}^*}$ is necessarily everywhere non-singular in Ω and the degree is either $+1$ or -1. However, the degree test does not use derivatives and can succeed even in cases where derivatives are either not at hand, or when the Jacobian matrix is potentially singular. For loop detection, this includes situations such as in Figure 21, where the self-crossing is close to parallel.

Similarly, the degree test can be shown to be more pow-

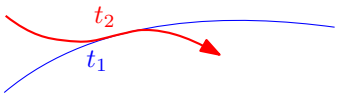


Figure 21: A "non-transversal" loop like this can easily be detected by the degree test, but methods requiring non-singular Jacobian matrix will fail to verify it.

erful than other interval-based verification tests, such as Mirranda's or Borsuk's test, due to the following result [26, Thm 6]:

Whenever a function $\mathbf{f}^*$ *has a* robust zero *(one that cannot be removed by arbitrary small perturbations), then it can be verified by the degree test, assuming that we have a sufficient subdivision and sufficiently precise interval-measurements.*

One could still argue that such arbitrary precise interval approximations are practically not at hand. Here we state another variant of the optimality of the degree, which is adapted to the setting of this paper:

Proposition. Let Ω, $[\mathbf{f}]$, $[\mathbf{t}]_j, [\mathbf{b}]_k$ be as in Theorem 1 and assume further that the degree $\deg(\mathbf{f}^*, \Omega) = 0$ and that the interior of Ω is connected. Then there exists a function $\mathbf{g} \in [\mathbf{f}]$ such that

- $\mathbf{0} \notin \mathbf{g}(\Omega)$;
- $\mathbf{g}([\mathbf{t}]_j) \subseteq [\mathbf{f}]([\mathbf{t}]_j)$ for all j, and
- $\mathbf{g}([\mathbf{b}]_k) \subseteq [\mathbf{f}]([\mathbf{b}]_k)$ for all k.

In other words, whenever we detect a zero degree on some set Ω with connected interior, then it is still possible that $\mathbf{f}^*$ has no zero: indeed, the unknown function $\mathbf{f}^*$ may be the function $\mathbf{g}$ from the theorem.

If we subdivided our domain more and obtained more data, our region Ω could split into more components — for example, Ω_1 with a degree 1, and Ω_2 with a degree -1. Each Ω_i would then provably contain a zero. However, based only on the above interval evaluations, we cannot conclude the existence of a zero.

In particular, if we cannot conclude a zero based on the degree test, we argue that *no other test* (such as Newton) would conclude it either, unless more data or finer interval evaluations are provided. The proof of this statement is elementary[9], but it requires some necessary definitions from topology, so we omit it here in order to keep the paper self-contained and readable for a wide audience. Our main message is to underline the usefulness of the degree test for zero detection of functions with bounded uncertainty, and its relevancy for loop closure proofs.

[9]The main idea is to define the function $\mathbf{g}$ to be equal to $\mathbf{f}^*$ on $\partial\Omega$ and, in a small enough ϵ-neighborhood of the boundary, to extend it to a positive scalar multiple of $\mathbf{f}^*$ such that its norm is small enough for any x that is ϵ-far from the boundary. This map takes $\{x : \operatorname{dist}(x, \partial\Omega) = \epsilon\}$ into a sphere of small diameter, and due to the fact that the degree is zero, can be extended to a function $\mathbf{g} : \Omega \to \mathbb{R}^n$ that it is still small farther from the boundary, and avoids zero.