Considering Adjacent Sets for Computing the Visibility Region*

Quentin Brateau[ab], Fabrice Le Bars[ac], and Luc Jaulin[ad]

Abstract

This paper explores the problem of the paving of the union of adjacent contractors. The focus is first put on the analysis of the topology of a set operator, which can be stable or not stable. Then, depending on the stability of the union operator, solutions are proposed to avoid fake boundaries in stable and non-stable union of sets. For stable unions of sets, a boundary preserving form will be developed to add a set overlapping the fake boundary in the expression of the union, whereas for non-stable union of sets, a boundary approach will be developed to avoid fake boundaries. Some problem-specific solutions are also developed to avoid fake boundaries. As an example, an enhancement of the separator on the visibility constraint is proposed. This avoids fake boundaries while characterizing the set of non-visible points from an observation point relative to a polygon.

Keywords: set methods, interval analysis, contractors, set inversion, topology

1 Introduction

Interval Arithmetic and contractor programming have emerged as powerful tools in the field of robotics [16, 5, 10, 8], offering robust methods for handling uncertainty and performing set-based computations. These techniques have been widely applied in robotics area such as in localization [5, 11], in path planning [3, 6], and in control of systems [19, 14].

Interval analysis is a subset of set methods where sets are represented by intervals. Some operators are defined in classical set theory, such as union, intersection, complementary of sets and so on [17]. As intervals are representing sets, these operators are also defined for intervals by interval arithmetic [8].

*This work has been supported by the French Government Defense procurement and technology agency (AID)

[a]LabSTICC UMR 6285, Robex team, ENSTA Bretagne, 2 rue Francois Verny, 29200, Brest, France

[b]E-mail: quentin.brateau@ensta-bretagne.org, ORCID: 0000-0002-1553-9549

[c]E-mail: fabrice.le_bars@ensta-bretagne.fr, ORCID: 0000-0001-9413-4621

[d]E-mail: lucjaulin@gmail.com, ORCID: 0000-0002-0938-0615

DOI: 10.14232/actacyb.314640

Contractors are a mathematical function acting on intervals. They are used to contract a domain of feasible values relative to a constraint. Denoting by $\mathbb{IR}^n$ the set of axis-aligned boxes of $\mathbb{R}^n$, the operator $\mathcal{C} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ is a contractor for $\mathbb{X} \subseteq \mathbb{R}^n$ if it meets the condition Equation (1).

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathcal{C}([\mathbf{x}]) \subseteq [\mathbf{x}] & (Contractance) \\ \mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} & (Completeness) \end{cases} \tag{1}$$

Figure 1 shows a graphical representation of a contractor. The contractor $\mathcal{C}_{\mathbb{A}}$ is contracting the provided box $[\mathbf{b}]$ relative to the set $\mathbb{A}$ in green. The resulting box is shown in purple and represent the contracted box $\mathcal{C}_{\mathbb{A}}([\mathbf{b}])$. To meet the conditions of Equation (1), the contractor is only able to remove points outside the set $\mathbb{A}$.

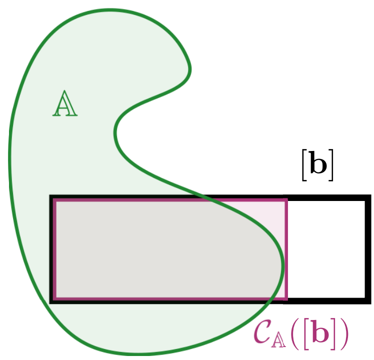


Figure 1: Graphical representation of a contractor

A contractor represents a set as defined in [2]. Set operators are then defined for contractors as it is for sets. Contractors can be combined by computing their union, their intersection, their cartesian product, and so on.

However, while most of these operations are well-defined and straightforward to implement, the union operation presents unique challenges, particularly when dealing with non-overlapping sets. The union of adjacent sets (i.e. sets that share a common boundary) using contractors can sometimes result in the appearance of fake boundaries at the interface of these sets. This phenomenon, was first highlighted in [20], and a solution was proposed using appropriate Disjunctive Normal Forms (DNF) and Conjunctive Normal Forms (CNF) [1] to avoid these fake boundaries. However, this solution is not always applicable as sets are not always defined as unions and intersections of sets.

For instance, the set of visible points from an observation point relative to an obstacle is defined and implemented in [4]. By paving this set, fake boundaries may appear in the non-visible area. Figure 2 shows an example of the set of visible points from an observation point relative to a polygon obstacle. The observation point is shown in red. The set of visible points from this observation point is shown in blue, the set of non-visible points is shown in pink, and the set of uncertain points is shown in yellow. The obstacle is outlined in black. Fake boundaries appear within the pink area, as lines of yellow boxes in the non-visible area.

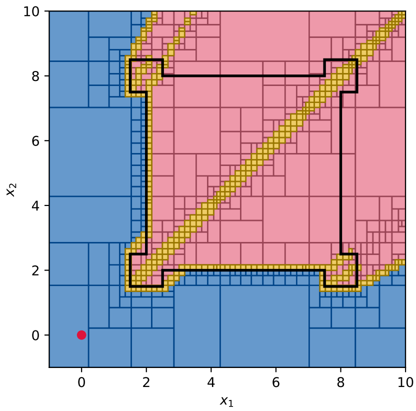


Figure 2: Separator on the visibility constraint

This article aims to address the problem of union operations on adjacent contractors, with a focus on eliminating fake boundaries. By developing new techniques for performing union operations on adjacent sets, we seek to minimize the added pessimism to the results, to improve the efficiency of the paving algorithm, and to enhance the accuracy and reliability of set-based computations in robotic applications. Added pessimism comes from bad classified boxes around the fake boundary. These uncertain boxes might suggest a boundary, but they are clearly inside the considered set.

This paper is organized as follows. Section 2 present the problem of the union of adjacent contractors by an introducing example. Then, Section 3 analyze the problem from a topological point of view, and distinguish stable and non-stable set operators. Section 4 and Section 5 present the solutions to avoid fake boundaries in both cases. Section 6 presents an application of the boundary approach on the separator on the visibility constraint. Finally, Section 7 concludes the paper.

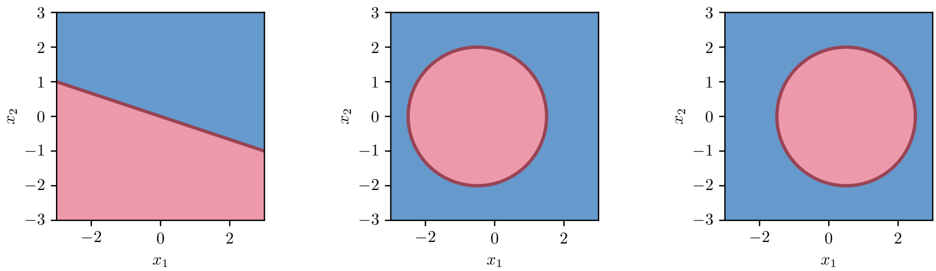
2 Problem Statement

2.1 Illustrative example

Consider three sets $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$ defined below (2):

$$\begin{aligned} \mathbb{A} &: \{x_1 + 3 \cdot x_2 \in [-\infty, 0]\} \\ \mathbb{B} &: \{(x_1 + 0.5)^2 + x_2^2 \in [-\infty, 4]\} \\ \mathbb{C} &: \{(x_1 - 0.5)^2 + x_2 \in [-\infty, 4]\} \end{aligned} \tag{2}$$

These sets are shown in Figure 3. The interior of the set is shown in pink, and the exterior is shown in blue.



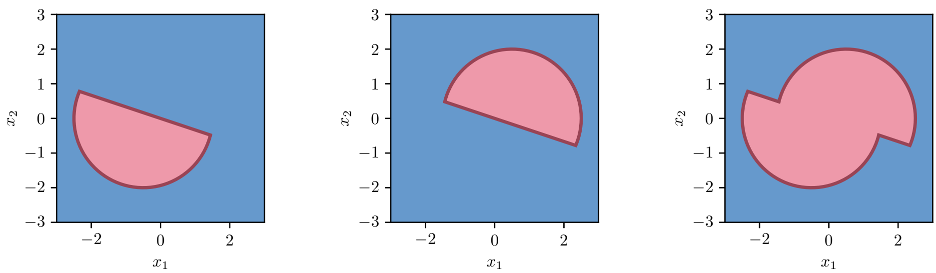
(a) Set $\mathbb{A}$ (b) Set $\mathbb{B}$ (c) Set $\mathbb{C}$

Figure 3: Sets $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$

Define a set $\mathbb{Z}$, computed using $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$, as shown in Equation (3).

$$\mathbb{Z} = (\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C}) \tag{3}$$

Set $\mathbb{Z}$, shown in Figure 4c, is built as the union of sets $\mathbb{A} \cap \mathbb{B}$ and $\overline{\mathbb{A}} \cap \mathbb{C}$ represented in Figure 4a, and Figure 4b.



(a) Set $\mathbb{A} \cap \mathbb{B}$ (b) Set $\overline{\mathbb{A}} \cap \mathbb{C}$ (c) $(\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C})$

Figure 4: Construction of set $\mathbb{Z}$ from $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$

Note that the two sets $\mathbb{A} \cap \mathbb{B}$ and $\overline{\mathbb{A}} \cap \mathbb{C}$ share a common and non-overlapping boundary. While paving set $\mathbb{Z}$ using the SIVIA algorithm [8], this common boundary appears as shown in Figure 5. This boundary is called a fake boundary [20] as it is not supposed to belong to $\mathbb{Z}$.

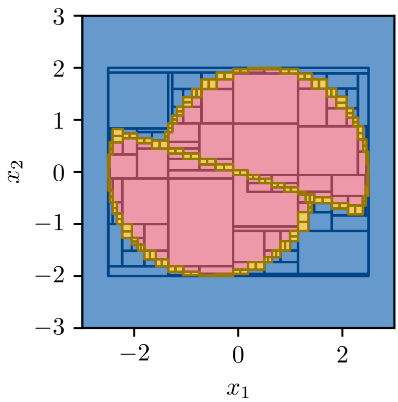


Figure 5: Paving of set $\mathbb{Z}$

2.2 Paving point of view

As shown in Figure 6, the paving algorithm is unable to classify an inner box $[\mathbf{b}]$ overlapping the fake boundary as fully inside $\mathbb{Z}$. Using contractors defined for $\mathbb{Z}$, inner parts $[\mathbf{b}] \setminus [\mathbf{b}_1] = [\mathbf{b}] \setminus \mathcal{C}_{\mathbb{A}\cap\mathbb{B}}([\mathbf{b}])$, and $[\mathbf{b}] \setminus [\mathbf{b}_2] = [\mathbf{b}] \setminus \mathcal{C}_{\overline{\mathbb{A}}\cap\mathbb{C}}([\mathbf{b}])$ are well classified. The remaining part $[\mathbf{b}_3] = [\mathbf{b}] \setminus [\mathbf{b}_1] \setminus [\mathbf{b}_2]$ is classified as unknown and is bisected until the paving algorithm reaches the desired precision.

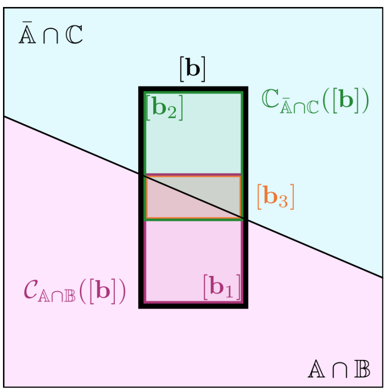
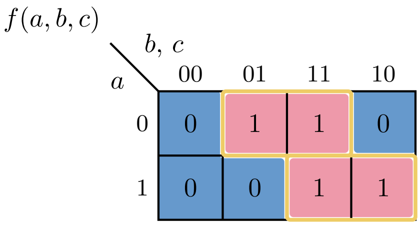


Figure 6: Paving of the fake boundary

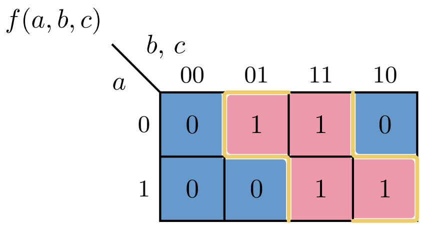
To avoid this issue, the paving algorithm has to take into account the fact that $\mathbb{A} \cup \overline{\mathbb{A}} = \mathbb{R}^n$. With this piece of information, the box $[\mathbf{b}]$ can be classified as fully inside $\mathbb{Z}$ in one step.

2.3 Karnaugh map point of view

Karnaugh maps for $(\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C})$ and $\mathbb{Z}$ are respectively shown in Figure 7a and Figure 7b. The interior is shown in pink, the exterior is shown in blue, and the boundary is shown in yellow. Although the interior and the exterior of these two sets are equal, the boundaries differ. By denoting by $\delta\mathbb{A}$ the boundary of a set $\mathbb{A}$, the fake boundary appearing on the paving in Figure 5 is $\partial\mathbb{A} \cap \mathbb{B} \cap \mathbb{C}$ and is exactly the difference between the boundaries of $(\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C})$ and $\mathbb{Z}$.



(a) Karnaugh map of $(\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap C)$



(b) Karnaugh map of $\mathbb{Z}$

Figure 7: Comparing Karnaugh maps of $(\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C})$ and $\mathbb{Z}$

2.4 Raised issues

Fake boundaries raise two issues. First they add pessimism to the results by classifying boxes around the common boundary as uncertain, whereas these boxes clearly belong to the union of the two sets. Secondly, fake boundaries slow down the paving algorithm by causing unnecessary box bisections around them.

3 Stability of Set Operators

3.1 Topological analysis of set operators

To better understand the issue around the union of adjacent sets, we need to define some tools to analyze the origin of these fake boundaries. Actually, although it turns out that fake boundaries may occur regardless of whether a set operator is Hausdorff-stable, solutions to avoid these fake boundaries are not the same in the two cases.

3.2 Hausdorff distance

Let $(\mathbb{S}, d)$ be a metric space. Define the ϵ-fattening [12] of a set $\mathbb{X}$ of $\mathbb{S}$ by Equation (4).

$$\mathbb{X}_\epsilon = \bigcup_{x \in \mathbb{X}} \{z \in \mathbb{S} \mid d(z, x) \leq \epsilon\} \tag{4}$$

The ϵ-fattening of a set $\mathbb{X}$ is the set of all points in $\mathbb{S}$ that are at most ϵ away from a point in $\mathbb{X}$ relative to the distance d of the metric space, as shown in Figure 8.

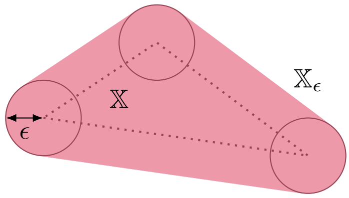


Figure 8: ϵ-fattening of a set

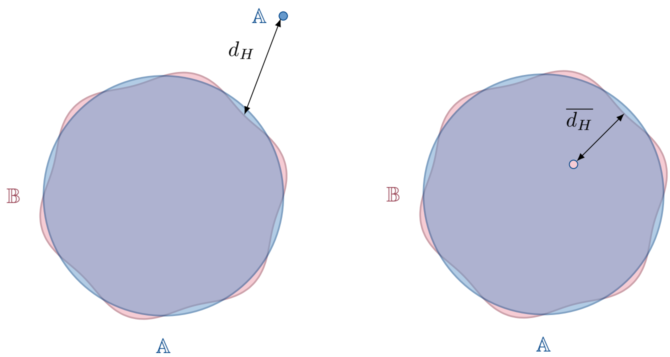
The Hausdorff distance [12] between two subsets $\mathbb{X}$ and $\mathbb{Y}$ of $\mathbb{S}$ is defined by Equation (5):

$$d_H(\mathbb{X}, \mathbb{Y}) = inf\{\epsilon \in \mathbb{R}^+ \mid \mathbb{X} \subseteq \mathbb{Y}_\epsilon \quad and \quad \mathbb{Y} \subseteq \mathbb{X}_\epsilon\} \tag{5}$$

We also introduce the complementary Hausdorff distance defined in Equation (6):

$$\overline{d_H}(\mathbb{X}, \mathbb{Y}) = d_H(\overline{\mathbb{X}}, \overline{\mathbb{Y}}) \tag{6}$$

Example. Figure 9 illustrate cases where Hausdorff distance and complementary Hausdorff distance are significant. Figure 9a shows an example of two sets $\mathbb{A}$ and $\mathbb{B}$ with $d_H(\mathbb{A}, \mathbb{B})$ large because of the small part of $\mathbb{A}$ far from the main part, but $\overline{d_H}(\mathbb{A}, \mathbb{B})$ is tiny, whereas Figure 9b shows an example where $d_H(\mathbb{A}, \mathbb{B})$ is tiny and $\overline{d_H}(\mathbb{A}, \mathbb{B})$ is large because of the hole in $\mathbb{A}$.



(a) Large $d_H(\mathbb{A}, \mathbb{B})$ (b) Large $\overline{d_H}(\mathbb{A}, \mathbb{B})$

Figure 9: Illustration of large Hausdorff and complementary Hausdorff distances

To take into account the general topology of sets, and to be able to compare it, the generalized Hausdorff distance is introduced and defined in Equation (7). It is the maximum between the Hausdorff distance and the complementary Hausdorff distance.

$$H_d(\mathbb{X}, \mathbb{Y}) = max\{d_H(\mathbb{X}, \mathbb{Y}), \overline{d_H}(\mathbb{X}, \mathbb{Y})\} \tag{7}$$

3.3 Hausdorff stability

Consider two subsets $\mathbb{X}$ and $\mathbb{Y}$ of $\mathbb{S}$. Then a binary operator $\diamond$ acting on set $\mathbb{X}$ and $\mathbb{Y}$ is stable if it meets condition of Equation (8).

$$\forall \eta \in \mathbb{R}_+, \quad \exists \epsilon \in \mathbb{R}_+^*, \quad \begin{cases} H_d(\mathbb{X}, \widetilde{\mathbb{X}}) & \leq \epsilon \\ H_d(\mathbb{Y}, \widetilde{\mathbb{Y}}) & \leq \epsilon \end{cases} \implies H_d(\mathbb{X} \diamond \mathbb{Y}, \widetilde{\mathbb{X}} \diamond \widetilde{\mathbb{Y}}) \leq \eta \tag{8}$$

Example. Consider two subsets $\mathbb{A}$ and $\mathbb{B}$ shown in Figure 10a and two other sets $\widetilde{\mathbb{A}}$ and $\widetilde{\mathbb{B}}$ shown in Figure 10b.

For the union operator, $d_H(\mathbb{A} \cup \mathbb{B}, \widetilde{\mathbb{A}} \cup \widetilde{\mathbb{B}})$ is small, but $\overline{d_H}(\mathbb{A} \cup \mathbb{B}, \widetilde{\mathbb{A}} \cup \widetilde{\mathbb{B}})$ is large as the union of $\widetilde{\mathbb{A}}$ and $\widetilde{\mathbb{B}}$ generates holes at the common boundary of $\mathbb{A}$ and $\mathbb{B}$. Then $H_d(\mathbb{A} \cup \mathbb{B}, \widetilde{\mathbb{A}} \cup \widetilde{\mathbb{B}})$ is large, and the union operator is not Hausdorff-stable for these sets, as it does not meet the condition of Equation (8).

Example. Consider two sets $\mathbb{A}$ and $\mathbb{B}$ shown in Figure 10a and two other sets $\widetilde{\mathbb{A}}$ and $\widetilde{\mathbb{B}}$ shown in Figure 10b.

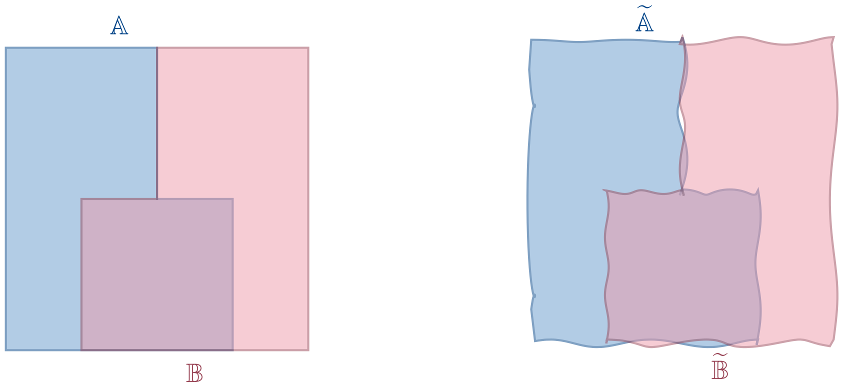
For the intersection operator, $\overline{d_H}(\mathbb{A} \cap \mathbb{B}, \widetilde{\mathbb{A}} \cap \widetilde{\mathbb{B}})$ is small, but $d_H(\mathbb{A} \cap \mathbb{B}, \widetilde{\mathbb{A}} \cap \widetilde{\mathbb{B}})$ is large as the intersection of $\widetilde{\mathbb{A}}$ and $\widetilde{\mathbb{B}}$ generates residual sets at the common boundary of $\mathbb{A}$ and $\mathbb{B}$. Then $H_d(\mathbb{A} \cap \mathbb{B}, \widetilde{\mathbb{A}} \cap \widetilde{\mathbb{B}})$ is large, and the intersection operator is not Hausdorff-stable for these sets, as it does not meet the condition of Equation (8).

Example. Consider the illustrative example presented in Section 2. The union operator between $\mathbb{A} \cap \mathbb{B}$ and $\overline{\mathbb{A}} \cap \mathbb{C}$ is Hausdorff stable as the generalized Hausdorff distance is small. This come from the fact that the same set $\mathbb{A}$ is involved in the computation of $H_d(\mathbb{A} \cap \mathbb{B}, \widetilde{\mathbb{A}} \cap \widetilde{\mathbb{B}})$ and $H_d(\overline{\mathbb{A}} \cap \mathbb{C}, \widetilde{\overline{\mathbb{A}}} \cap \widetilde{\mathbb{C}})$.

This Hausdorff stability condition characterizes the fact that a small perturbation on sets will change the topology of the result by opening boundaries or creating additional ones. It allows identifying topologically different problems. Adapted solutions for Hausdorff-stable and non Hausdorff-stable problems will be proposed in the following sections.

4 Stable Case Solution: Boundary-Preserving Form

In the Hausdorff-stable case, it is possible to change the expression of the computed set $\mathbb{Z}$ by adding a set overlapping the fake boundary. This set helps in the classification of boxes around the fake boundary in the paving algorithm. It must



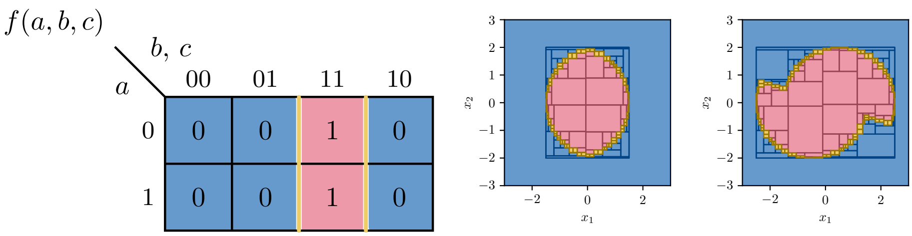
(a) Sets $\mathbb{A}$ and $\mathbb{B}$ (b) Sets $\widetilde{\mathbb{A}}$ and $\widetilde{\mathbb{B}}$

Figure 10: $\mathbb{A}$ and $\mathbb{B}$ are not Hausdorff-stable for union and intersection operators

be chosen such that the interior and the exterior of $\mathbb{Z}$ are preserved, but also its boundary. In this example, the set $\mathbb{D} = \mathbb{B} \cap \mathbb{C}$ is added to the expression of $\mathbb{Z}$ which becomes $\mathbb{Z}'$ Equation (9).

$$\mathbb{Z}' = (\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C}) \cup (\mathbb{B} \cap \mathbb{C}) \tag{9}$$

The Karnaugh map of the set $\mathbb{D}$ is shown in Figure 11a, and the paving of $\mathbb{D}$ is shown in Figure 11b. This set ensures that the Karnaugh map of $\mathbb{Z}'$ is the same as the Karnaugh map of $\mathbb{Z}$ shown in Figure 7b. The resulting paving of $\mathbb{Z}'$ is shown in Figure 11c. There is no more fake boundaries.



(a) Karnaugh map of $\mathbb{D}$ (b) Set $\mathbb{D}$ (c) Set $\mathbb{Z}'$

Figure 11: Boundary preserving form

Using the boundary preserving form leads to a correct paving without any fake boundaries. Therefore, to use this solution, the set boundaries have to be analyzed to find the fake boundaries and add a set overlapping these fake boundaries in the expression of the set to pave. This approach is working but is problem-specific and needs to be adapted on a case-by-case basis. This method works well in the Hausdorff-stable case, as there is the possibility to add a set that overlap the fake boundary. For non Hausdorff-stable operators, the boundary preserving form is not possible as the Karnaugh map is not highlighting any sets that can be added to the expression of the paved set to avoid fake boundaries, and another approach is needed.

5 Non-Stable Case: Boundary Approach

5.1 Topology of the boundary

Let $T = (\mathbb{S}, \tau)$ be a topological space. $\forall \mathbb{X} \in \mathbb{S}$, denote by $\overline{\mathbb{X}}$ the complementary of $\mathbb{X}$ in $\mathbb{S}$, by $cl_{\mathbb{S}}(\mathbb{X})$ the closure of $\mathbb{X}$ in $\mathbb{S}$, by $int_{\mathbb{S}}(\mathbb{X})$ the interior of $\mathbb{X}$ in $\mathbb{S}$, and by $\partial\mathbb{X}$ the boundary of $\mathbb{X}$ in $\mathbb{S}$.

Theorem. *Let $T = (\mathbb{S}, \tau)$ be a topological space. Then*

$$\forall(\mathbb{A}, \mathbb{B}) \in \mathbb{S}^2 \quad \partial(\mathbb{A} \cup \mathbb{B}) \subseteq \partial\mathbb{A} \cup \partial\mathbb{B}$$

Proof. By definition of the boundary

$$\forall \mathbb{A} \in \mathbb{S}, \quad \partial\mathbb{A} = cl_{\mathbb{S}}(\overline{\mathbb{A}}) \cap cl_{\mathbb{S}}(\mathbb{A})$$

By property, intersection is a subset of each set

$$\forall(\mathbb{A}, \mathbb{B}) \in \mathbb{S}^2, \quad \begin{cases} \mathbb{A} \cap \mathbb{B} \subseteq \mathbb{A} \\ \mathbb{A} \cap \mathbb{B} \subseteq \mathbb{B} \end{cases}$$

Then

$$\begin{aligned} \partial(\mathbb{A} \cup \mathbb{B}) &= cl_{\mathbb{S}}(\overline{\mathbb{A} \cup \mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{A} \cup \mathbb{B}) \\ &= cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{A} \cup \mathbb{B}) \\ &= cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap (cl_{\mathbb{S}}(\mathbb{A}) \cup cl_{\mathbb{S}}(\mathbb{B})) \\ &= (cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{A})) \cup cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap cl_{S}(\mathbb{B}) \\ &\subseteq (cl_{\mathbb{S}}(\overline{\mathbb{A}}) \cap cl_{\mathbb{S}}(\mathbb{A})) \cup (cl_{\mathbb{S}}(\overline{\mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{B})) \\ &= \partial\mathbb{A} \cup \partial\mathbb{B} \end{aligned}$$ □

Section 5.1 demonstrates that the boundary is not preserved over union of sets as $\partial(\mathbb{A} \cup \mathbb{B}) \subseteq \partial\mathbb{A} \cup \partial\mathbb{B}$. This is why the paving of the union of contractors leads to fake boundaries.

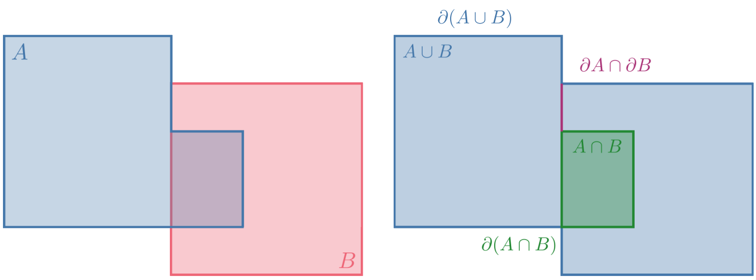
Section 5.1 present the general formula for the union of the boundary of two sets.

Theorem. *Let (S, τ) be a topological space. Then*

$$\forall(\mathbb{A}, \mathbb{B}) \in \mathbb{S}^2 \qquad \partial\mathbb{A} \cup \partial\mathbb{B} = \partial(\mathbb{A} \cup \mathbb{B}) \cup \partial(\mathbb{A} \cap \mathbb{B}) \cup (\partial\mathbb{A} \cap \partial\mathbb{B})$$

Proof. Section 5.1 is proven in [9]. □

From Section 5.1, it is noticeable that the union of boundaries is not the boundary of union. This is the reason why $\mathbb{Z}$ and $(\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C})$ do not have the same boundaries when paving these sets. An illustration of Section 5.1 is shown in Figure 12.



(a) Set $\mathbb{A}$ and $\mathbb{B}$ (b) Decomposition of $\partial\mathbb{A} \cup \partial\mathbb{B}$

Figure 12: Illustration of Section 5.1

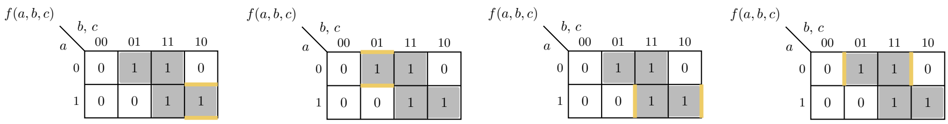
Remark. When $\mathbb{A} \cap \mathbb{B} = \emptyset$ and $\partial\mathbb{A} \cap \partial\mathbb{B} = \emptyset$ in Section 5.1, the union of boundaries is the boundary of union. This is the case where the sets are non-overlapping with no common boundary.

5.2 Boundary approach

A boundary approach can be used to get rid of this fake boundary. This will help to solve this purely computational problem, as the mathematical expression of the set $\mathbb{Z}$ do not have any fake boundaries. This approach consists in computing the boundary of the set $\mathbb{Z}$. This boundary will separate an inner and an outer subpaving. The classification of the resulting subpavings as inside or outside is done using a predicate. The boundary approach method was first introduced in [7] to speed up the solving of set inversion problems.

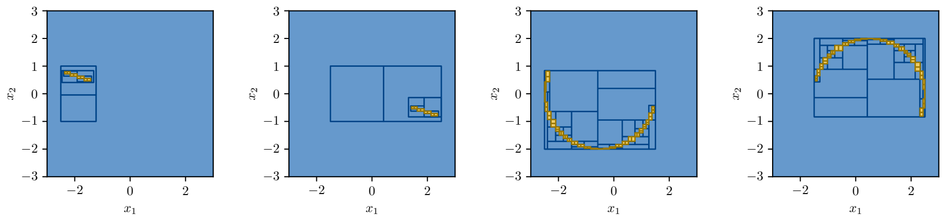
First, $\partial\mathbb{Z}$ has to be expressed from set $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$ without the fake boundary. Figure 13 and Figure 14 respectively show Karnaugh maps and paving of intermediate sets involved in the building of $\partial\mathbb{Z}$. Then, $\partial\mathbb{Z}$ is computed as the union of these boundaries, and it matches the Karnaugh map of $\mathbb{Z}$ shown in Figure 7b.

Then using a predicate, the connected subsets separated by $\partial\mathbb{Z}$ are classified as inside or outside. This predicate is based on the expression of $\mathbb{Z}$ of Equation (3), and is tested on box corners until an in and an out points are found. Then, boxes containing each point are classified as in and out boxes, and the information



(a) $\partial\mathbb{A} \cap \mathbb{B} \cap \overline{\mathbb{C}}$ (b) $\partial\mathbb{A} \cap \overline{\mathbb{B}} \cap \mathbb{C}$ (c) $\mathbb{A} \cap \partial\mathbb{B}$ (d) $\overline{\mathbb{A}} \cap \partial\mathbb{C}$

Figure 13: Karnaugh map of the boundaries

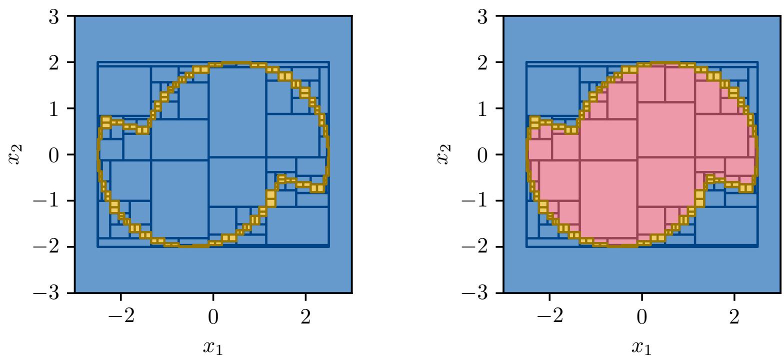


(a) $\partial\mathbb{A} \cap \mathbb{B} \cap \overline{\mathbb{C}}$ (b) $\partial\mathbb{A} \cap \overline{\mathbb{B}} \cap \mathbb{C}$ (c) $\mathbb{A} \cap \partial\mathbb{B}$ (d) $\overline{\mathbb{A}} \cap \partial\mathbb{C}$

Figure 14: Building the boundary of $\mathbb{Z}$

is propagated near to near without crossing the boundary. Finally, each box is classified as in, out, or uncertain.

Figure 15a shows $\partial\mathbb{Z}$ built from boundaries shown in Figure 14 using the proposed method. The resulting paving of Z is shown in Figure 15b, which is classified using the subpaving coloration method.



(a) ∂Z (b) Z

Figure 15: Boundary approach

This boundary approach is efficient to get rid of fake boundaries. Set $\mathbb{Z}$ is computed from the union of two separators, $\mathcal{S}_{\mathbb{A}\cap\mathbb{B}}$, and $\mathcal{S}_{\overline{\mathbb{A}}\cap\mathbb{C}}$, and this union is reinforced by a contractor on the boundary $\mathcal{C}_{\partial\mathbb{Z}}$.

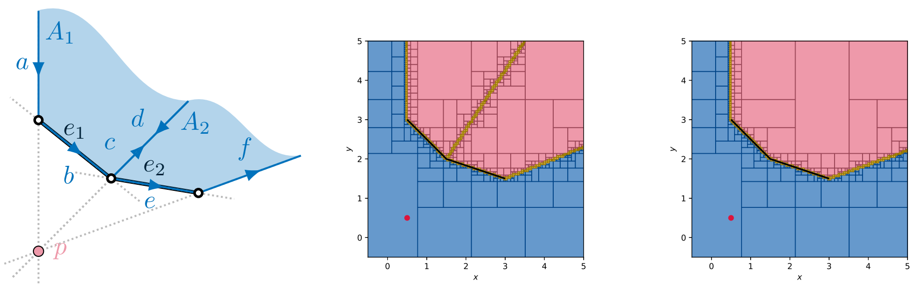
Remark. This method is also working for the Hausdorff-stable case, but it is more efficient to use the boundary preserving form presented in Section 4, as the contractor on the boundary is not easy to define, and the subpaving coloration method is not needed.

6 Application

6.1 Boundary approach application to the separator on the visibility constraint

Separator over the visibility constraint, as implemented in [4], suffer from this fake boundaries when it deals with polygon obstacles. In fact, the contractor on the visibility constraint is defined for an obstacle segment. The extension to polygons involves the union of non-visible areas relative to each segment, and this union leads to fake boundaries.

Figure 16a shows an illustration of the separator on the visibility constraint as implemented in [4]. For each obstacle segment, three segments are defined, which separate the visible and non-visible parts of the space. For segment e_1 is defined relative to the observation point p, the oriented half space on the left of segment a, the one on the left of segment b, and the same for segment c. It is the same for the set of visible points for segment e_2 defined by half planes on the left of segments d, e, and f. The set of masked points from p by e_1 and e_2 i then the union of these two sets A_1 and A_2. Paving this separator shows a fake boundary as shown in Figure 16b.



(a) Separator construction (b) Classical approach (c) Boundary approach

Figure 16: Separator on the visibility constraint using the boundary approach

To avoid this problem, the boundary approach can be applied. The set of masked points from observation point p relative to segments e_1 and e_2 should be defined by half planes on the left of segments a, b, e, and f. The simplification of $c = -d$ has to be taken into account while contracting to avoid this fake boundary. This simplification is based on algebraic topology [18] in which boundary simplifications are defined and used. Figure 16c shows the paving of this separator using the boundary approach. There is no longer fake boundaries appearing.

Remark. For now, fake boundaries have to be identified and removed by hand, as it not the main topic of this paper. Neither [4] nor this work propose an automatic boundary simplification to avoid fake boundaries in union of adjacent sets. Therefore, it is necessary to find solutions that are problem-specific in order to avoid fake boundaries, as developed in the next subsection.

6.2 Toward a generic implementation of the separator on the visibility constraint

In the case of the visibility constraint another approach to solve this problem can be proposed. The set of visible points from an observation point placed at $(0, 0)$ relative to a shape $\mathbb{Y}$ can be defined by :

$$\mathbb{S} = \{x \in \mathbb{R}^2, \exists \alpha \in \mathbb{R} \mid \alpha \cdot \mathbf{x} \in \mathbb{Y}\} \tag{10}$$

Denoting by f the homothety defined in Equation (11).

$$\begin{aligned} f : \mathbb{R}^3 &\mapsto \mathbb{R}^2 \\ (\mathbf{x}, \alpha) &\to \alpha \cdot \mathbf{x} \end{aligned} \tag{11}$$

Remark. If the observation point is not placed at (0, 0), a simple translation of the problem leads to the presented solution.

The set $\mathbb{S}$ can then be defined as the projection of $f(\mathbb{Y})$ for $\alpha \in [0, 1]$. Listing 1 show the implementation of this separator using the Codac Library [15]. Figure 17a shows the paving of this implementation of the visibility constraint. The comparison with Figure 17b, where the classical implementation of this constraint from [4] on the same obstacle polygon is shown, validates that the problem of fake boundaries is avoided with this method. Figure 17a requires 391 bisections whereas Figure 17b requires 321 bisections. The complexity of these two approaches is quite similar, although all the tests carried out lead to a slightly higher number of bisections for the proposed method, with the benefit of a set without fake boundaries. This is mainly due to the projection algorithm which induces bisections in the dimension of the homothety factor α.

Figure 18 shows the paving of the visibility separator on the same obstacle presented in Figure 2, but the proposed method avoid fake boundaries.

Remark. The separator representing the obstacle could be any separator. However, the separator must be in a closed form with an interior. This method is not applicable to segments or open polygons for instance. But the advantage of this approach is that it can be applied to an ellipsis obstacle.

Remark. There are polygons for which the separator on the visibility constraint does not generate fake boundaries. In these cases the classical implementation proposed in [4] is more efficient than the proposed approach, as the algorithm used to project a separator is based on contractors over quantifiers which requires bisections [2, 13].

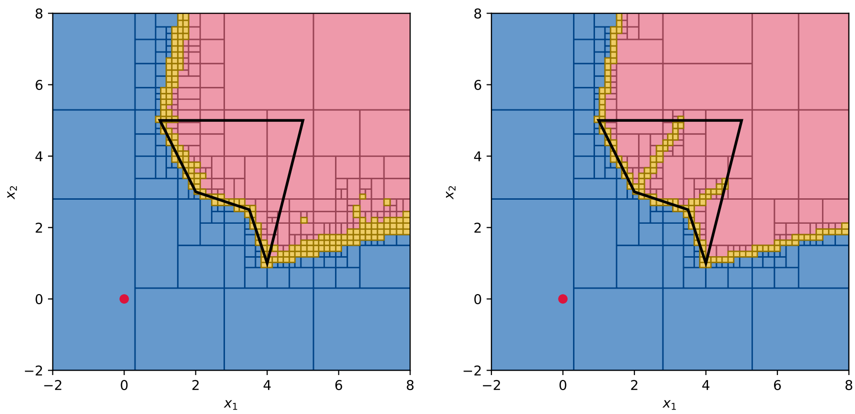
```python
import codac as cd

# Set Y definition
polygon = [[2, 3], [3.5, 2.5], [4, -1], [5, 5], [1, 5], [2, 3]]
Sy = cd.SepPolygon(polygon)

# Set Z definition
f = cd.Function("x", "y", "a", "(a*x,a*y)")
Sz = cd.SepInverse(Sy, f)

# Projection of for a in [0, 1]
epsilon = 0.1
Sx = cd.SepProj(Sz, cd.Interval(0, 1), epsilon)
```

Listing 1: Separator on the visibility constraint using Codac Library



(a) Proposed implementation (b) Classical implementation

Figure 17: Generic SepVisible implementation

Figure 19 shows the comparison between the classical and the projection approaches on the paving of a visibility separator without fake boundaries. The proposed implementation shown in Figure 19a requires 419 bisections whereas the classical implementation shown in Figure 19b requires only 278 bisections.

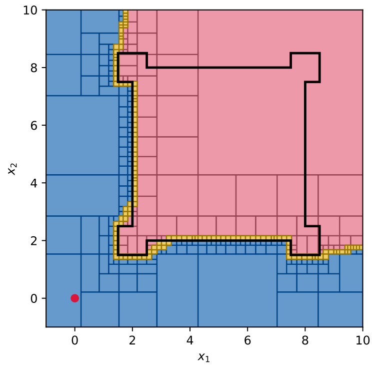
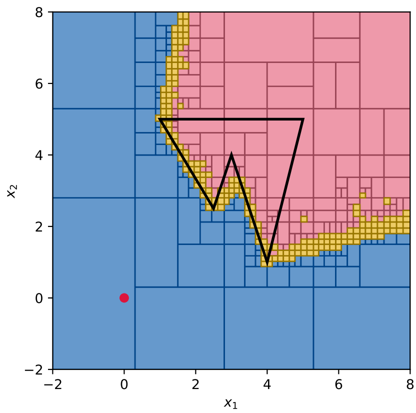
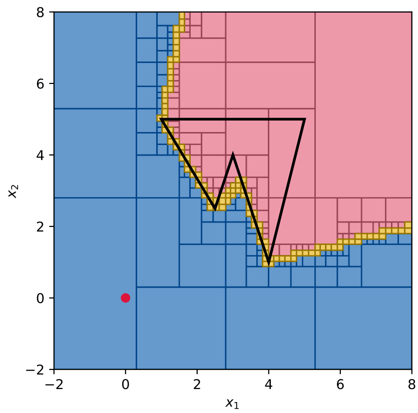


Figure 18: Separator on the visibility constraint on a room



(a) Proposed implementation



(b) Classical implementation

Figure 19: Generic implementation of the separator on the visibility constraint

7 Conclusion

In conclusion, this work has highlighted the problem associated with the union of adjacent contractors. Paving the union of these contractors creates fake boundaries that add pessimism to the results and increase the computation time.

This problem occurs in two cases: when an operator applied to sets is Hausdorff-stable, and when it is non-Hausdorff-stable. An approach for Hausdorff-stable is to use a boundary preserving form by adding sets overlapping the fake boundary in the expression of the paved set. For non-Hausdorff-stable operators, a boundary approach is proposed to get rid of this fake boundary.

The result shows that both these approaches are efficient in fake boundary avoidance. The drawback of these methods is that they are problem-specific, they need to be tuned for each problem, and this paper does not provide an automatic way to remove fake boundaries.

Finally, a generic implementation of the separator on the visibility constraint was proposed. This approach shows that sometimes the fake boundary problem can also be avoided by expressing the problem differently. There are, nevertheless, cases in which no false boundary appears in the classical implementation of the separator on the visibility constraint. In these cases, the classical implementation is more efficient than the proposed method, since the latter involves the projection of a separator, which is computationally time-consuming.

References

[1] Andrews, P. B. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Applied Logic Series. Springer Netherlands, 2013. DOI: 10.1007/978-94-015-9934-4.

[2] Chabert, G. and Jaulin, L. Contractor programming. *Artificial Intelligence*, 173:1079–1100, 2009. DOI: 10.1016/j.artint.2009.03.002.

[3] Chauhan, A., Vyas, P., Vachhani, L., and Maity, A. Optimal path planning for a non-holonomic robot using interval analysis. In *Proceedings of the Indian Control Conference*, pages 184–189, 2018. DOI: 10.1109/INDIANCC.2018.8307975.

[4] Guyonneau, R. *Méthodes Ensemblistes Pour La Localisation En Robotique Mobile*. PhD Thesis, Angers, 2013. URL: https://theses.hal.science/tel-00961501.

[5] Guyonneau, R., Lagrange, S., Hardouin, L., and Lucidarme, P. Guaranteed interval analysis localization for mobile robots. *Advanced Robotics*, 28(16):1067–1077, 2014. DOI: 10.1080/01691864.2014.908742.

[6] Jaulin, L. Path planning using intervals and graphs. *Reliable Computing*, 7(1):1–15, 2001. DOI: 10.1023/A:1011400431065.

[7] Jaulin, L. A boundary approach for set inversion. *Engineering Applications of Artificial Intelligence*, 100:104184, 2021. DOI: 10.1016/j.engappai.2021.104184.

[8] Jaulin, L., Kieffer, M., Didrit, O., Walter, E., Jaulin, L., Kieffer, M., Didrit, O., and Walter, É. *Interval analysis*. Springer, 2001. DOI: 10.1007/978-1-4471-0249-6.

[9] Law Wysocki, M. and Darmochwa, A. Subsets of topological spaces. *Journal of Formalized Mathematics*, 1, 1989. URL: https://mizar.uwb.edu.pl/JFM/Vol1/tops_1.html.

[10] Merlet, J.-P. Interval analysis and robotics. In Kaneko, M. and Nakamura, Y., editors, *Robotics Research*, pages 147–156, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-14743-2_13.

[11] Mourad, F., Snoussi, H., Abdallah, F., and Richard, C. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, 57(8):3226–3239, 2009. DOI: 10.1109/TSP.2009.2020018.

[12] Munkres, J. *Topology*. Featured Titles for Topology. Prentice Hall, Incorporated, 2000. ISBN: 9780131816299.

[13] Ninin, J. Global optimization based on contractor programming: An overview of the IBEX library. In Kotsireas, I. S., Rump, S. M., and Yap, C. K., editors, *Mathematical Aspects of Computer and Information Sciences*, pages 555–559, Cham, 2016. Springer International Publishing. DOI: 10.1007/978-3-319-32859-1_47.

[14] Rauh, A. and Hofer, E. P. Interval methods for optimal control. In *Variational Analysis and Aerospace Engineering*, pages 397–418, New York, NY, 2009. Springer New York. DOI: 10.1007/978-0-387-95857-6_22.

[15] Rohou, S., Desrochers, B., and Le Bars, F. The codac library. *Acta Cybernetica*, 26(4):881–887, 2024. DOI: 10.14232/actacyb.302772.

[16] Seignez, E., Kieffer, M., Lambert, A., Walter, E., and Maurin, T. Experimental vehicle localization by bounded-error state estimation using interval analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1084–1089, 2005. DOI: 10.1109/IROS.2005.1545155.

[17] Stoll, R. R. *Set Theory and Logic*. Dover books on advanced mathematics. Dover Publications, 1979. ISBN: 9780486638294.

[18] Tom Dieck, T. *Algebraic topology*, Volume 8. European Mathematical Society, 2008. DOI: 10.4171/048.

[19] Vehí, J., Ferrer, I., and Ángel Sainz, M. A survey of applications of interval analysis to robust control. *IFAC Proceedings Volumes*, 35(1):389–400, 2002. DOI: 10.3182/20020721-6-ES-1901.00399.

[20] Welte, A., Jaulin, L., Ceberio, M., and Kreinovich, V. Avoiding fake boundaries in set interval computing. *Journal of Uncertain Systems*, 11(2):137 – 148, 2017. URL: https://ensta-bretagne.hal.science/hal-01698416/file/jusVol11No2paper07.pdf.