Characterization of the Area Explored by an Autonomous Robot

Maria Luiza Costa Vianna [1,2] Eric Goubault [1]
Luc Jaulin [2] Sylvie Putot [1]

[1] Laboratoire d'Informatique de l'École Polytechnique (LIX)

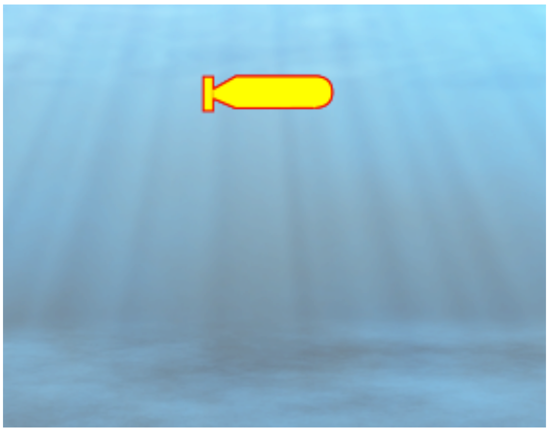
[2] ENSTA Bretagne, Lab-STICC

Palaiseau, 11/10/2021

Introduction

Autonomy

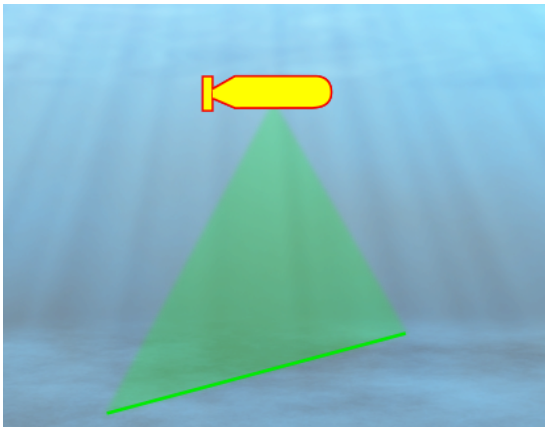
- Navigation sensors (Proprioceptive)
  - IMU, GPS, DVL . . .



Introduction

Autonomy

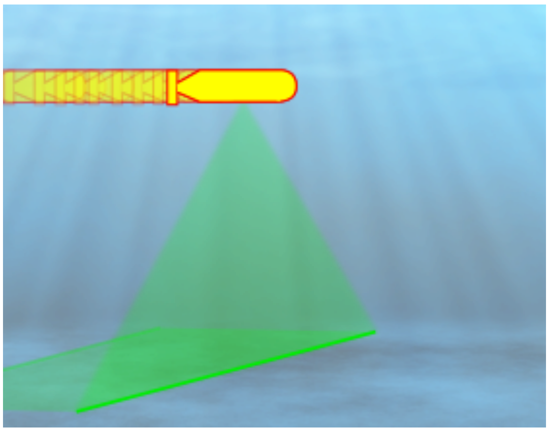
- Navigation sensors (Proprioceptive)
  - IMU, GPS, DVL ...
- Observation sensors (Exteroceptive)
  - Camera, sonar/lidar, temperature, salinity ...



Introduction

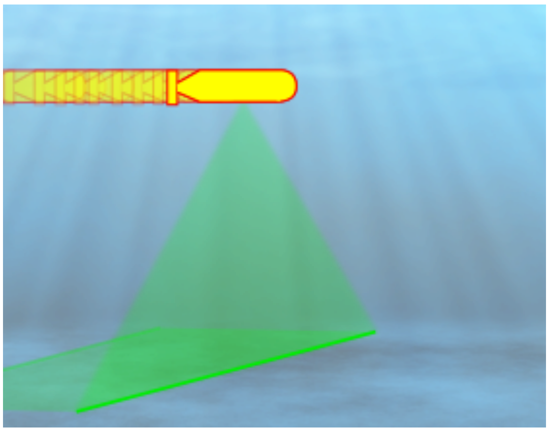
Explored Area

The explored area is the union of the visible areas over the whole trajectory.



Problem

- Compute the explored area.
- Compute the number of times each point in the environment has been explored.



Applications

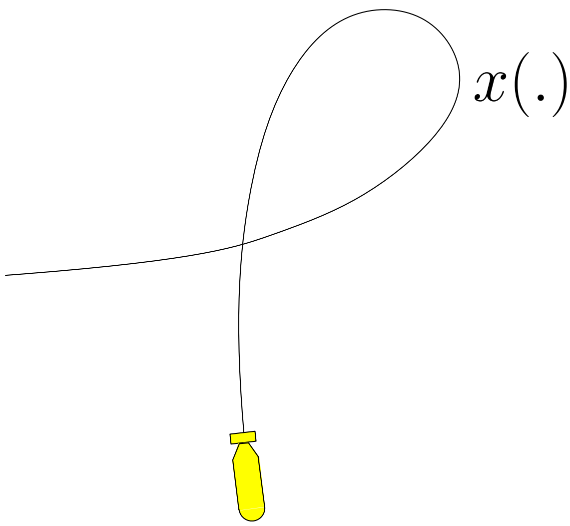
- Assess area-covering missions
  - Determine if a mission is complete
  - Plan other missions to fill possible gaps
- **Guarantee** that if a target is not detected, the target does not exist.

1. Introduction
2. **Problem Statement**
3. Problem Approach
4. Implementation
5. Results
6. Conclusions and Future Work

Problem Statement

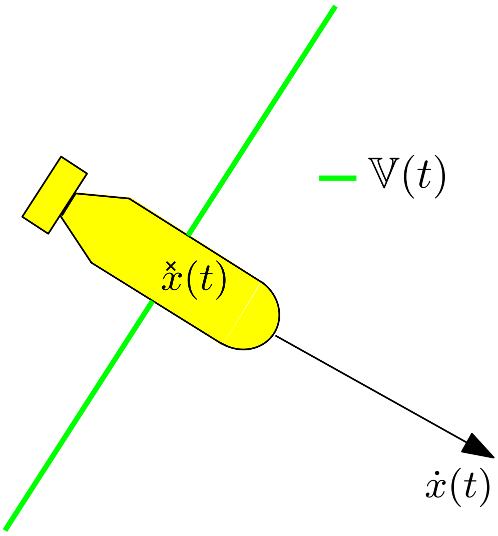
Hypothesis

- $x(.) : \mathbb{R} \to \mathbb{R}^2$
- $T = [0, T_{max}]$
- $x(.)$ is continuous in T.
- $x(.)$ and $\dot{x}(.)$ are known.



Problem Statement

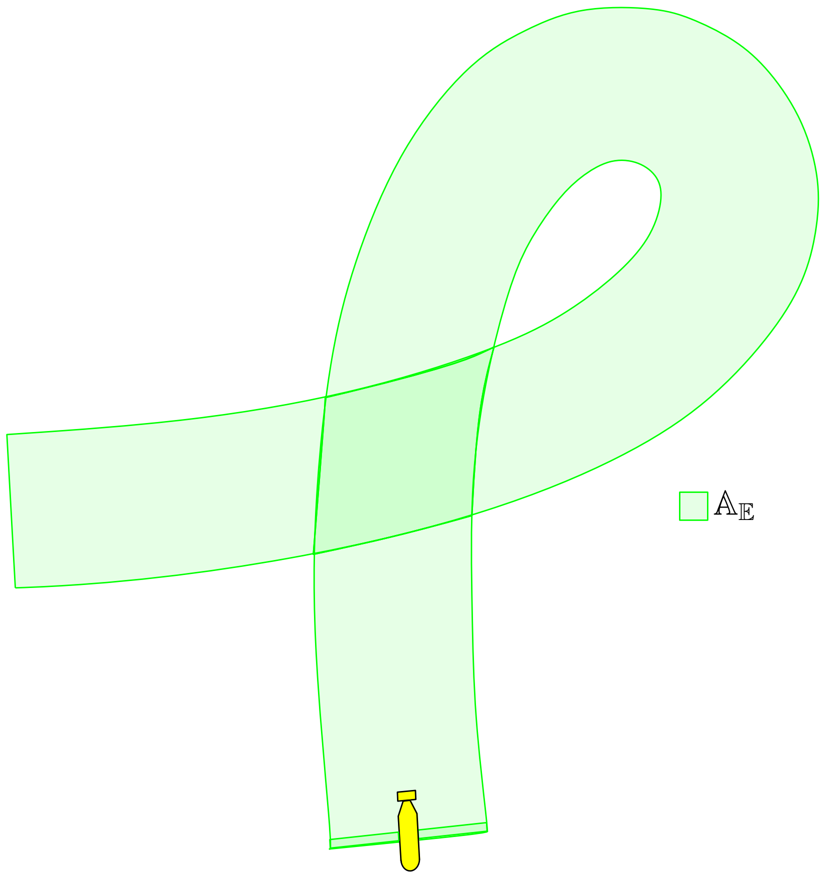
- $\mathbb{V}(x(t), \dot{x}(t))$ is the visible area at time t.



Problem Statement

$\mathbb{A}_{\mathbb{E}}$ is the explored area

$$\mathbb{A}_{\mathbb{E}} = \bigcup_{t \in T} \mathbb{V}(t)$$



Problem Statement

Entries

- $x(.)$, the robot's trajectory.
- $\dot{x}(.)$, the robot's trajectory derivatives.
- T, time interval.
- $\mathbb{V}(.)$, visible area

Problem Statement

Entries

- $x(.)$, the robot's trajectory.
- $\dot{x}(.)$, the robot's trajectory derivatives.
- T, time interval.
- $\mathbb{V}(.)$, visible area

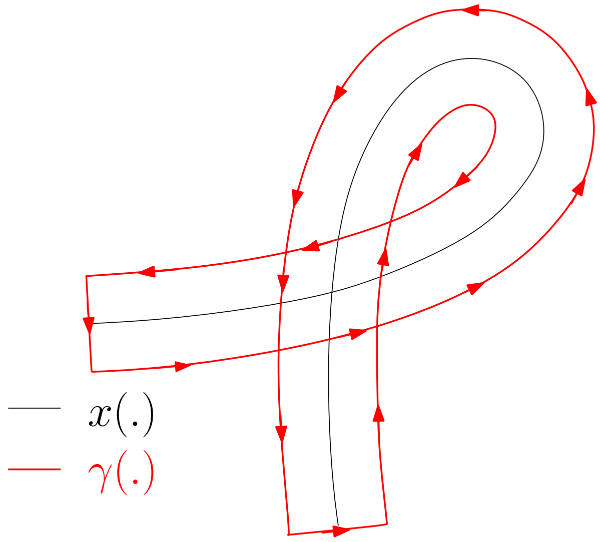
Desired Output

- Guaranteed approximation of the explored area $\mathbb{A}_{\mathbb{E}}$.
- Number of times each point in $\mathbb{A}_{\mathbb{E}}$ was seen during the mission.

Sonar Contour

From the robot's trajectory $x(.)$ and the knowledge of the range of visibility of each observation sensor, the sonar contour $\gamma(.)$ can be defined as illustrated.

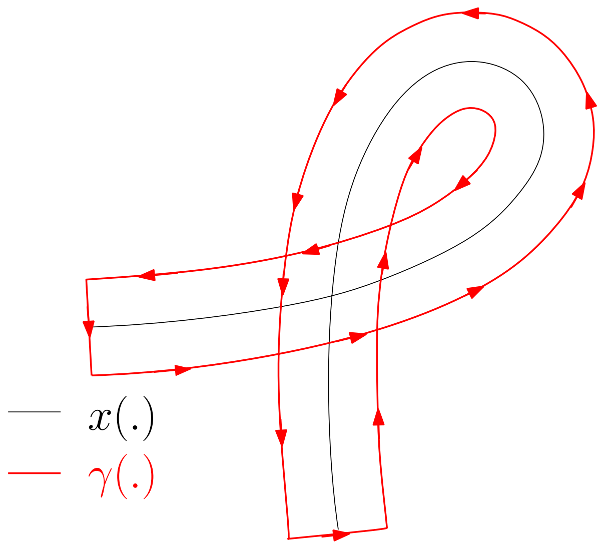
- $\gamma(.) : \mathbb{R} \rightarrow \mathbb{R}^2$
- $T_\gamma = [0, 1]$
- $\gamma(.)$ is continuous in T_γ.
- $\gamma(0) = \gamma(1)$.



Parameterization

- $x(.) : \mathbb{R} \to \mathbb{R}^2$
- $T = [0, T_{max}]$
- $x(.)$ is continuous in T.

- $\gamma(.) : \mathbb{R} \to \mathbb{R}^2$
- $T_\gamma = [0, 1]$
- $\gamma(.)$ is continuous in T_γ.
- $\gamma(0) = \gamma(1)$.



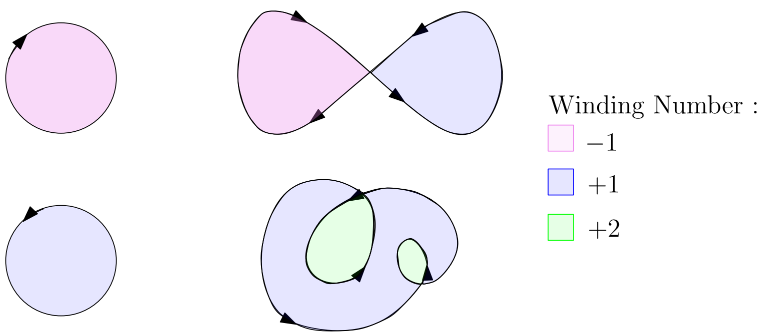
The parameterization of $x(.)$ and $\gamma(.)$ are not the same.

- $x(.)$ is parameterized by time $t \in T$.
- $\gamma(.)$ is parameterized by $\tau \in T_\gamma$.

Winding Number

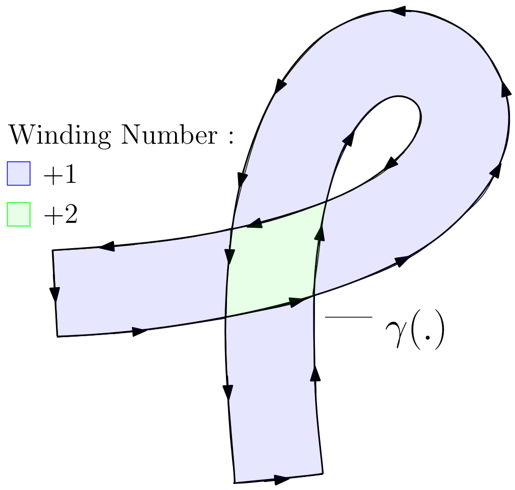
Winding Number

The winding number $\eta(\gamma(.), p)$ of a closed curve $\gamma(.)$ in the plane around a given point p is an integer representing the total number of times that curve travels counterclockwise around the point. [2]



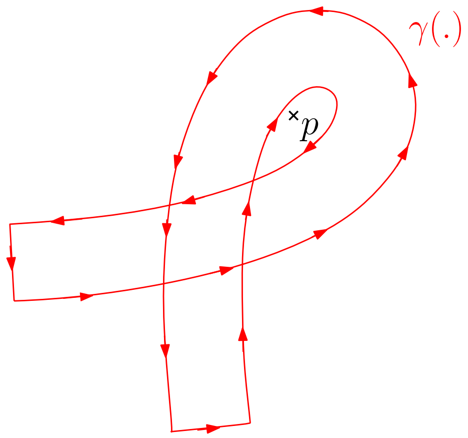
Problem Approach

$$\mathbb{A}_{\mathbb{E}} = \{z \in \mathbb{R}^2 | \eta(\gamma(.), z) \neq 0\}$$



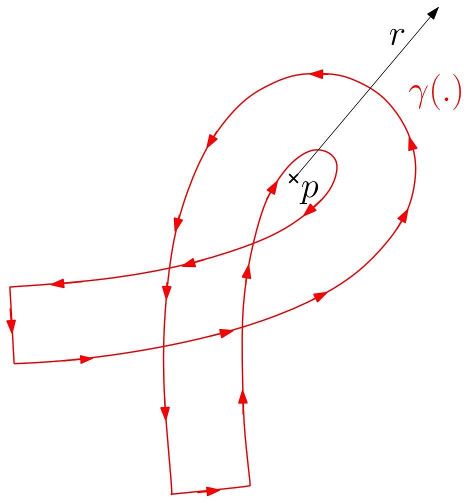
Winding Number Calculation

Given a closed contour $\gamma(.)$ and a point p, what is the winding number $\eta(\gamma(.), p)$?



Winding Number Calculation

Given a closed contour $\gamma(.)$ and a point p, what is the winding number $\eta(\gamma(.), p)$?

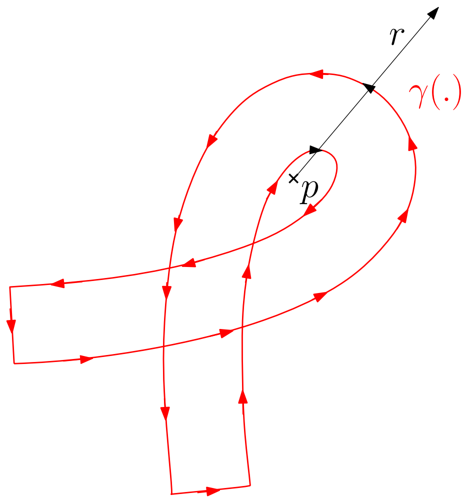


Dan Sunday's algorithm [4]

- Choose an infinite ray cast r from the point being checked .

Winding Number Calculation

Given a closed contour $\gamma(.)$ and a point p, what is the winding number $\eta(\gamma(.), p)$?

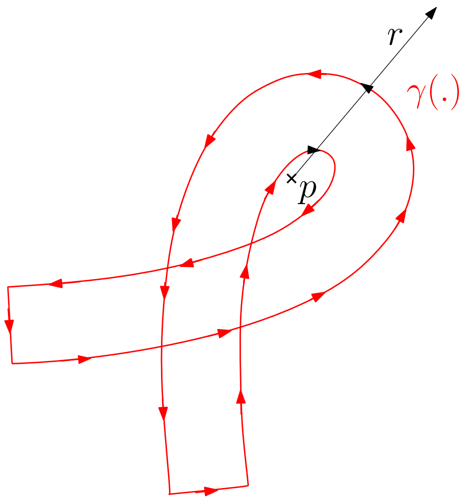


Dan Sunday's algorithm [4]

- Choose an infinite ray cast r from the point being checked .
- Identify intersections of r with $\gamma(.)$ and directions.

Winding Number Calculation

Given a closed contour $\gamma(.)$ and a point p, what is the winding number $\eta(\gamma(.), p)$?



$\eta(\gamma(.), p) = 1 - 1 = 0$

Dan Sunday's algorithm [4]

- Choose an infinite ray cast r from the point being checked .
- Identify intersections of r with $\gamma(.)$ and directions.
- Compute $\eta(\gamma(.), p)$

Problem Approach

Entries

- $\gamma(.)$, the sonar's contour.
- $\dot{\gamma}(.)$, the contour's derivatives.
- T_γ, contour's interval.

Problem Approach

Entries

- $\gamma(.)$, the sonar's contour.
- $\dot{\gamma}(.)$, the contour's derivatives.
- T_γ, contour's interval.

Desired Output

- Guaranteed approximation of the explored area $\mathbb{A}_\mathbb{E}$.
- Number of times each point in $\mathbb{A}_\mathbb{E}$ was seen during the mission.

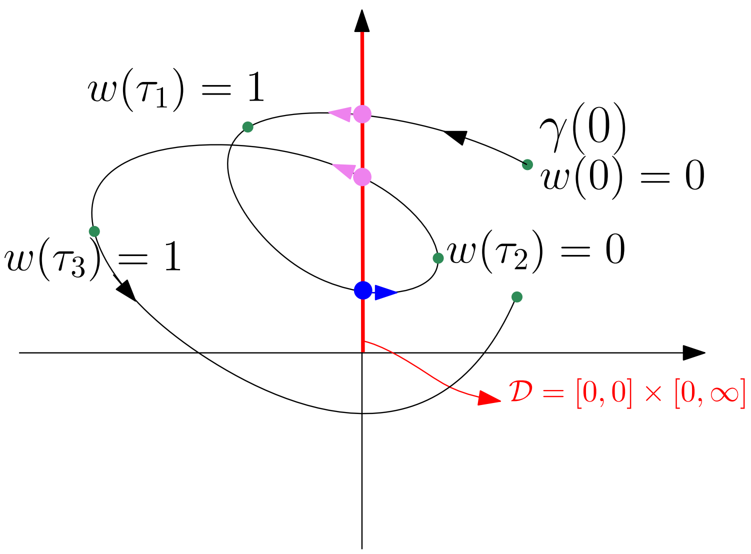
Proposed solution

$$\eta(\gamma(.), .)$$

Turning Number

- The objective of the turning number is to provide a sequential method for calculating the winding number w.r.t 0.
- $w(\tau) \in \mathbb{Z}$ is the turning number associated to $\gamma(\tau)$.

Turning Number

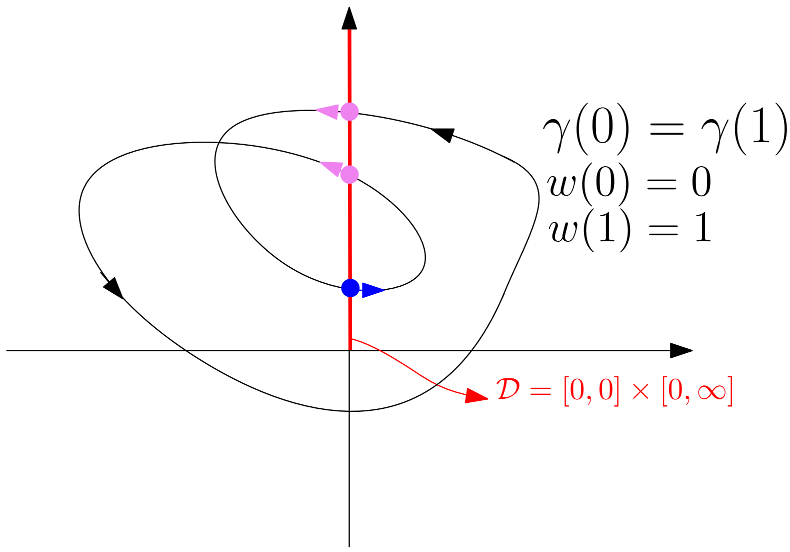


- $T^+(\tau) = \{t \in [0, \tau] \mid \gamma(t) \in \mathcal{D}$ and $\gamma_1(t - \delta) > 0\}$
- $T^-(\tau) = \{t \in [0, \tau] \mid \gamma(t) \in \mathcal{D}$ and $\gamma_1(t - \delta) < 0\}$
- $w(\tau) = \#T^+(\tau) - \#T^-(\tau)$

Problem Approach

The turning number allows a recursive calculation of the winding number.

$$\text{If } \gamma(0) = \gamma(1) \Rightarrow w(1) = \eta(\gamma(.), 0)$$



1. Introduction
2. Problem Statement
3. Problem Approach
4. **Implementation**
5. Results
6. Conclusions and Future Work

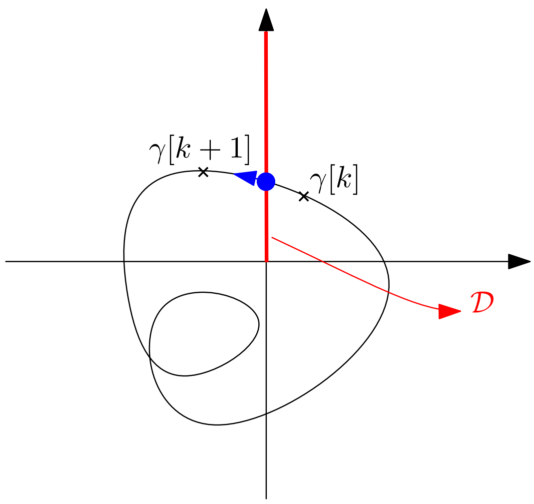
Implementation

Numerical Representation of the Contour

Numerical representation of the contour

Discretization

- $T_\gamma = [0, 1]$
- $\gamma(.)$ is continuous in T_γ.
- $\gamma(.)$ is evaluated at $\tau = \delta k$, where, $k = 0, 1, \ldots, \frac{1}{\delta}$ and $\delta \in \mathbb{R}$ is the discretization step.



Tubes

Definition

A tube is an envelope of trajectories, [3], [1] .

- $[\gamma](.) : \mathbb{R} \rightarrow \mathbb{IR}^n$
- $\gamma(.) \in [\gamma](.)$ if $\forall \tau \in T_\gamma$, $\gamma(\tau) \in [\gamma](\tau)$

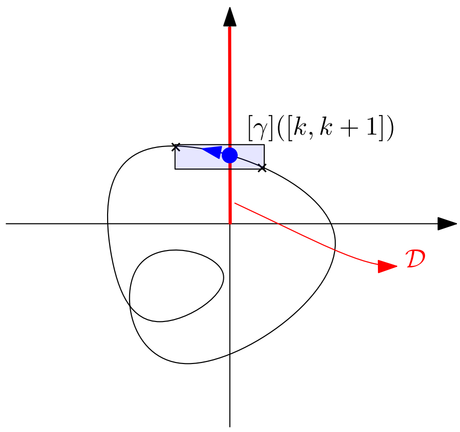
Tubes

Definition

A tube is an envelope of trajectories, [3], [1] .

- $[\gamma](.) : \mathbb{R} \rightarrow \mathbb{IR}^n$
- $\gamma(.) \in [\gamma](.)$ if $\forall \tau \in T_\gamma$, $\gamma(\tau) \in [\gamma](\tau)$

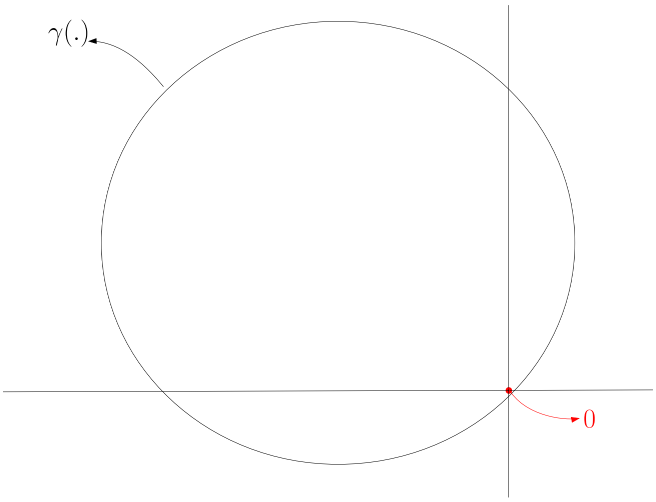
Example:
$[\gamma]([k, k+1])$ is the smallest box enclosing all solutions for $\gamma(\tau)$, such that, $\gamma(.) \in [\gamma](.)$ and $\tau \in [k\delta, (k+1)\delta]$.



Numerical representation of the contour

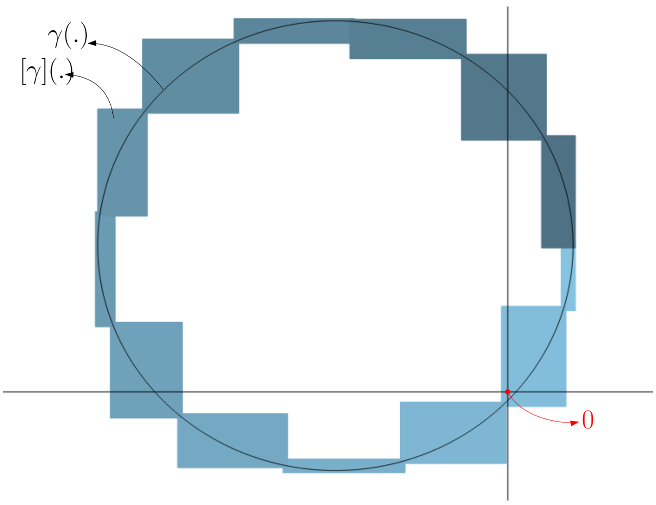
Example

- $\eta(\gamma(.), 0) \in \mathbb{Z}$



Numerical representation of the contour

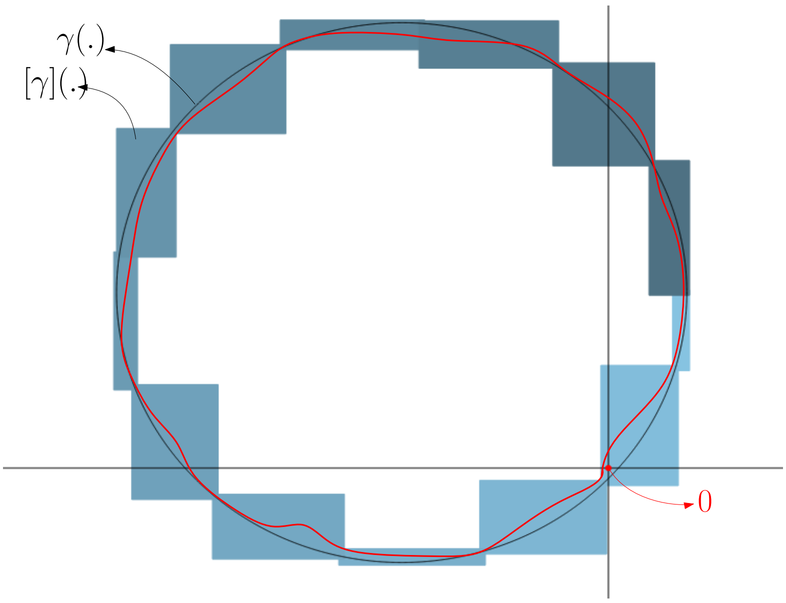
- $\gamma(.) \in [\gamma](.)$
- $\dot{\gamma}(.) \in [\dot{\gamma}](.)$
- $\eta([\gamma](.), 0) \in \mathbb{IZ}$
- $\eta(\gamma(.), 0) \in \eta([\gamma](.), 0)$



Numerical representation of the contour

Example

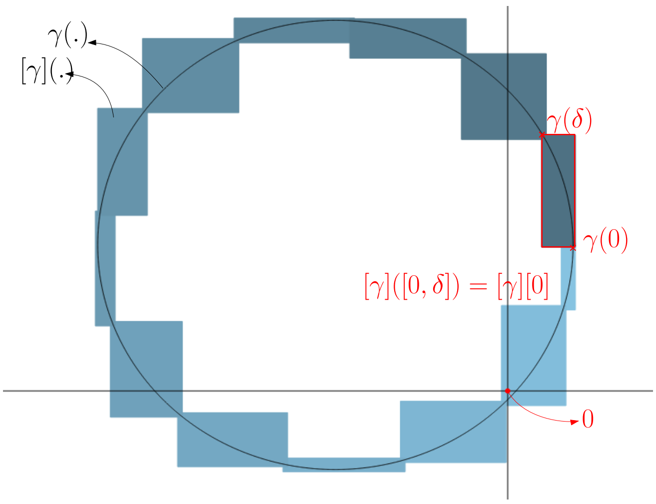
- $\eta([\gamma](.), 0) = [0, 1]$
- $\eta(\gamma(.), 0) = 1 \in \eta([\gamma](.), 0)$



Numerical representation of the contour

Discrete representation

$$[\gamma][k] = [\gamma]([k\delta, (k+1)\delta])$$
where, $k = 0, 1, \dots, \frac{1}{\delta} - 1$



Implementation
Algorithm

Algorithm

The winding number $\eta([\gamma](.), 0)$ can be sequentially computed using the turning number.

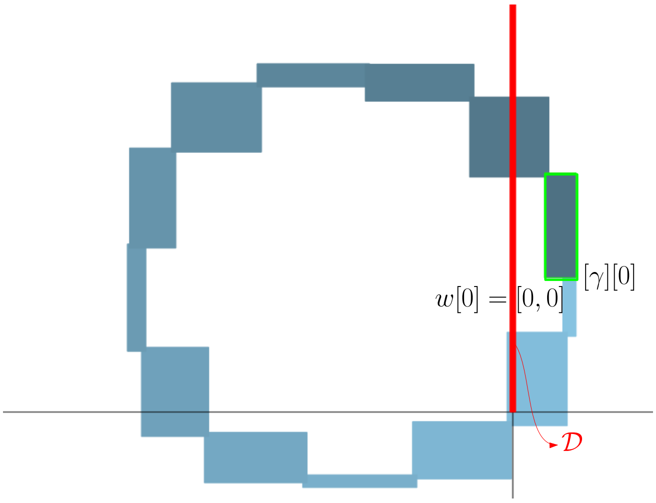
Algorithm

The winding number $\eta([\gamma](.), 0)$ can be sequentially computed using the turning number.

Initialization

- We assume that $[\gamma][0] \cap \mathcal{D} = \emptyset$

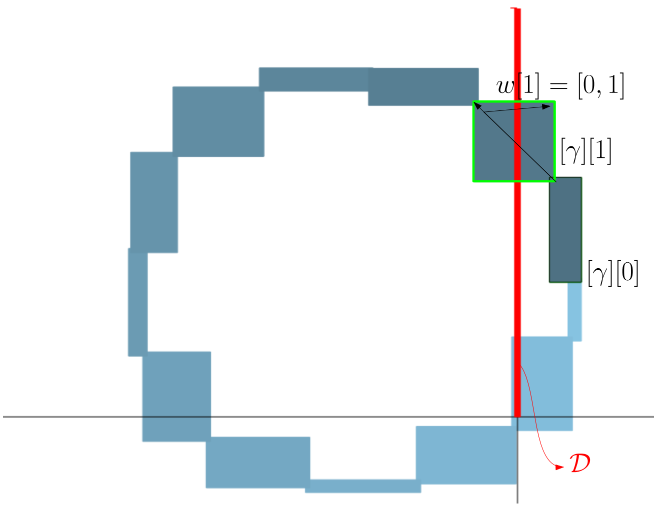
$$w[0] = [0, 0]$$



Algorithm

Rule 1

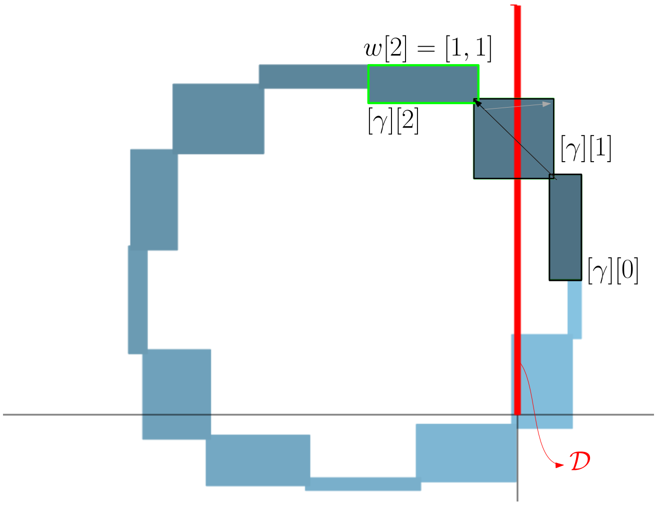
$$\left.\begin{array}{c} [\gamma][k] \cap \mathcal{D} \neq \emptyset \\ 0 \notin [\gamma][k] \\ [\gamma][k-1] \cap \mathcal{D} = \emptyset \\ [\gamma_1][k-1] \subset \mathbb{R}^+ \end{array}\right\} \quad w[k] = (w[k-1] + [0,1])$$



Algorithm

Rule 2

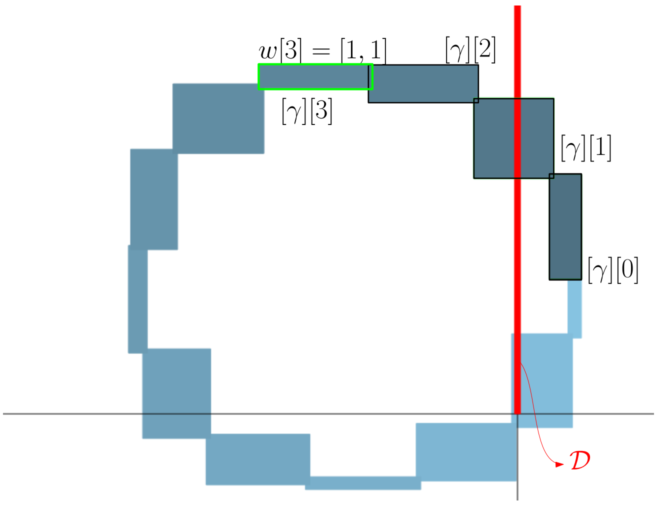
$$\left.\begin{array}{c}[\gamma][k]\cap\mathcal{D}=\emptyset\\ [\gamma_1][k]\subset\mathbb{R}^-\\ [\gamma][k-1]\cap\mathcal{D}\neq\emptyset\\ 0\notin[\gamma][k-1]\end{array}\right\}\quad [w][k]=[w^-[k-1]+1,w^+[k-1]]$$



Algorithm

Rule 3

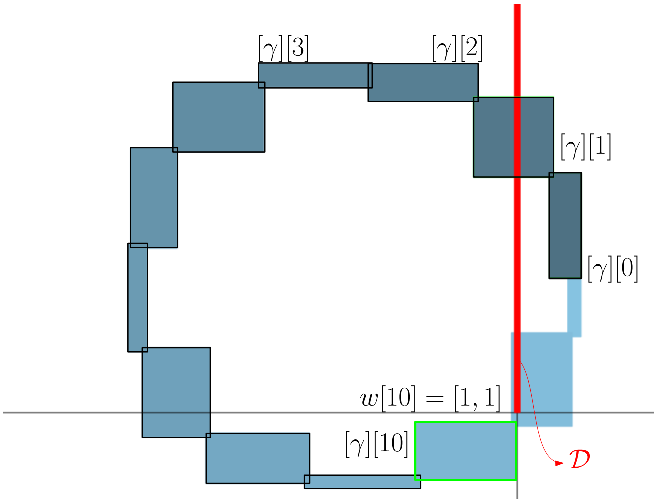
$$\left.\begin{array}{r}[x][k] \cap \mathcal{D} = \emptyset \\ [x][k-1] \cap \mathcal{D} = \emptyset\end{array}\right\} \quad [w][k] = [w][k-1]$$



Algorithm

Rule 3

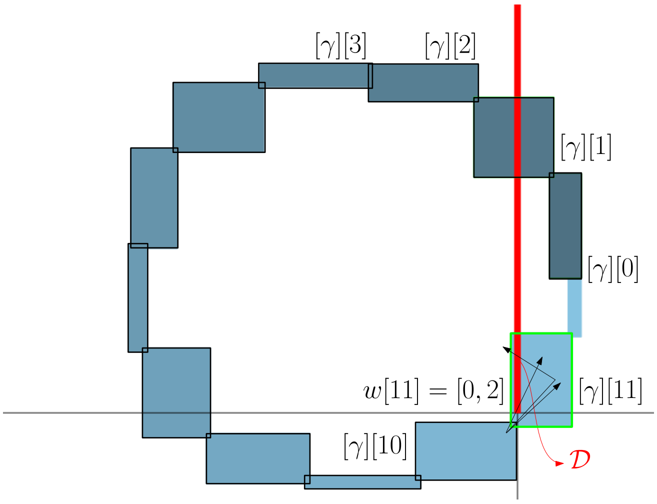
$$\left.\begin{array}{r}[x][k] \cap \mathcal{D} = \emptyset \\ [x][k-1] \cap \mathcal{D} = \emptyset\end{array}\right\} \quad [w][k] = [w][k-1]$$



Algorithm

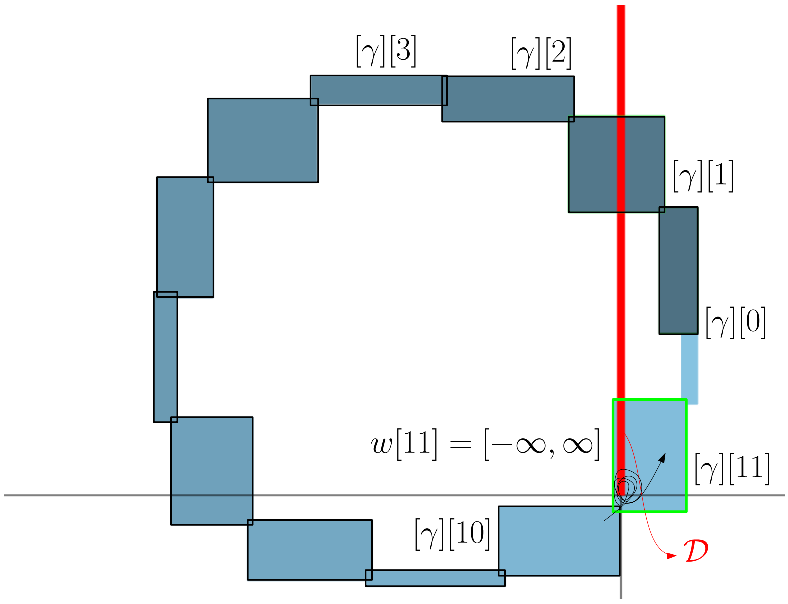
Rule 4

$$\left.\begin{array}{c}[\gamma][k-1]\cap\mathcal{D}=\emptyset\\ 0\in[\gamma][k]\\ 0\notin[\dot{\gamma}][k]\end{array}\right\}\quad [w][k]=[w][k-1]+[-1,1]$$



Algorithm

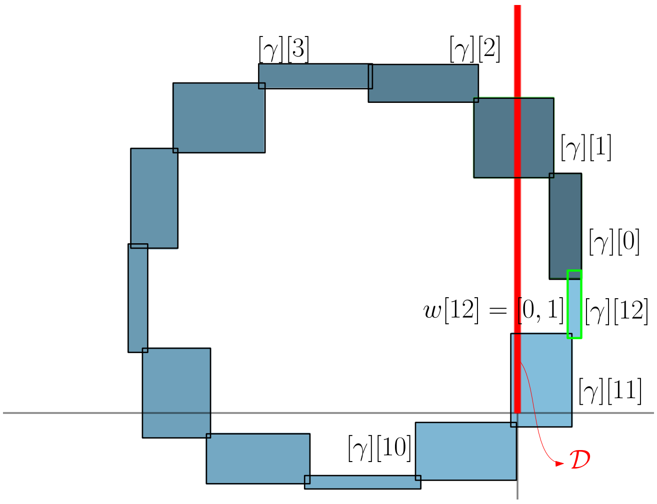
If $0 \in [\dot{\gamma}][k]$



Algorithm

Rule 5

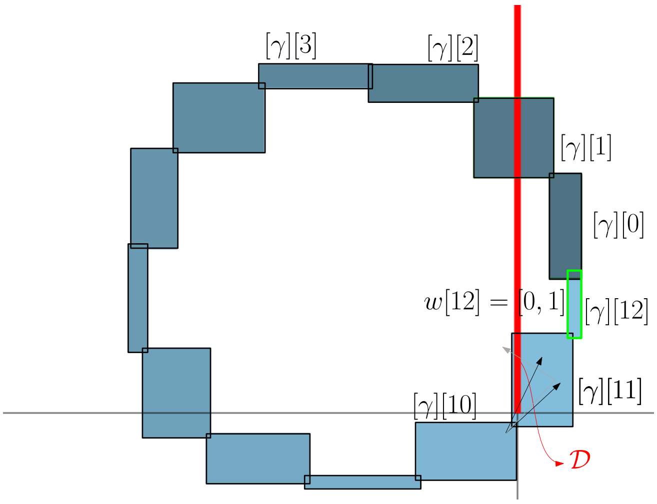
$$\left.\begin{array}{c}[\gamma][k] \cap \mathcal{D} = \emptyset \\ [\gamma_1][k] \subset \mathbb{R}^+ \\ 0 \in [\gamma][k-1]\end{array}\right\} \quad [w][k] = [w^-[k-1], w^+[k-1]-1]$$



Algorithm

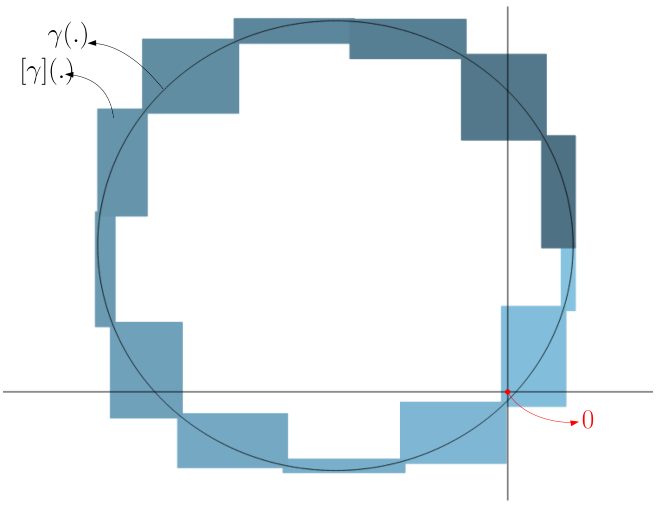
Rule 5

$$\left.\begin{array}{c}[\gamma][k]\cap\mathcal{D}=\emptyset\\ [\gamma_1][k]\subset\mathbb{R}^+\\ 0\in[\gamma][k-1]\end{array}\right\}\qquad [w][k]=[w^-[k-1],w^+[k-1]-1]$$



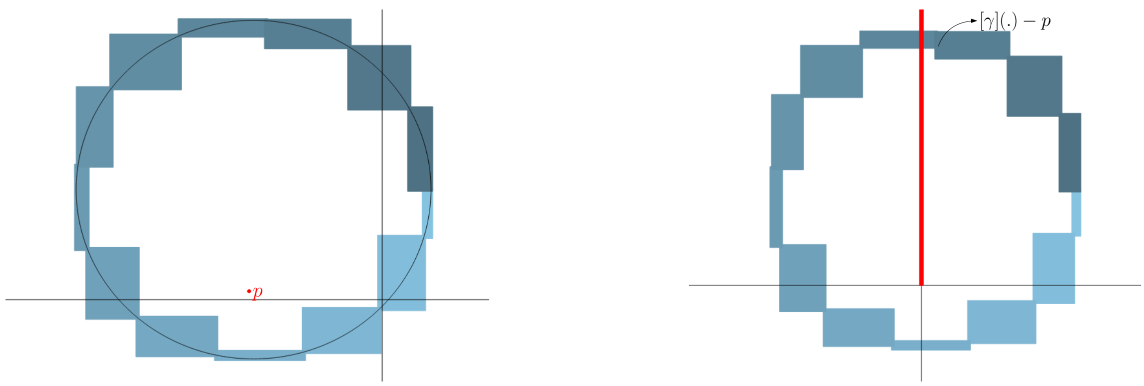
Algorithm

$$\eta([\gamma](.), 0) = [0, 1]$$
$$\eta(\gamma(.), 0) = 1 \in \eta([\gamma](.), 0)$$



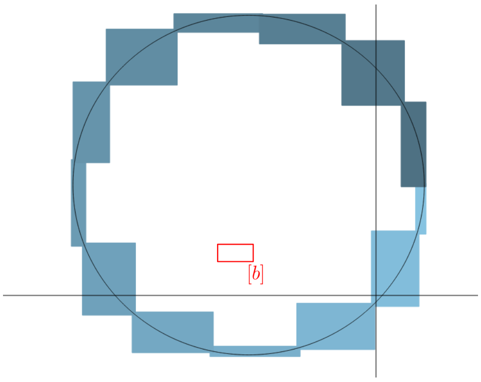
Algorithm

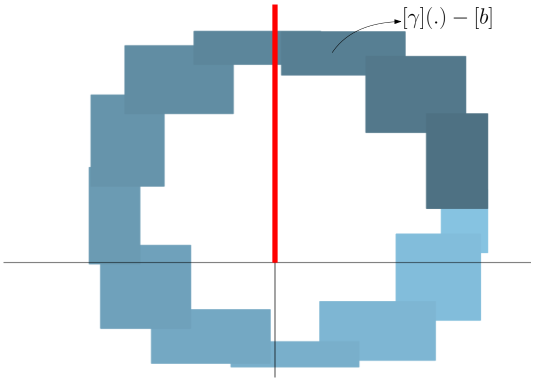
- $p \in \mathbb{R}^2$
  - $\eta([\gamma](.), p) = \eta([\gamma](.) - p, 0)$



Algorithm

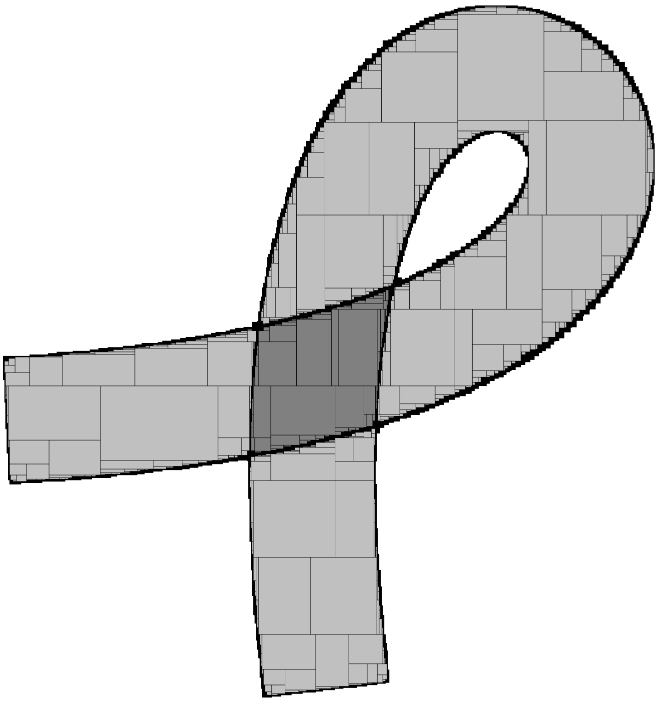
- $p \in \mathbb{R}^2$
  - $\eta([\gamma](.), p) = \eta([\gamma](.) - p, 0)$
- $[b] \in \mathbb{IR}^2$
  - $\eta([\gamma](.), [b]) = \eta([\gamma](.) - [b], 0)$



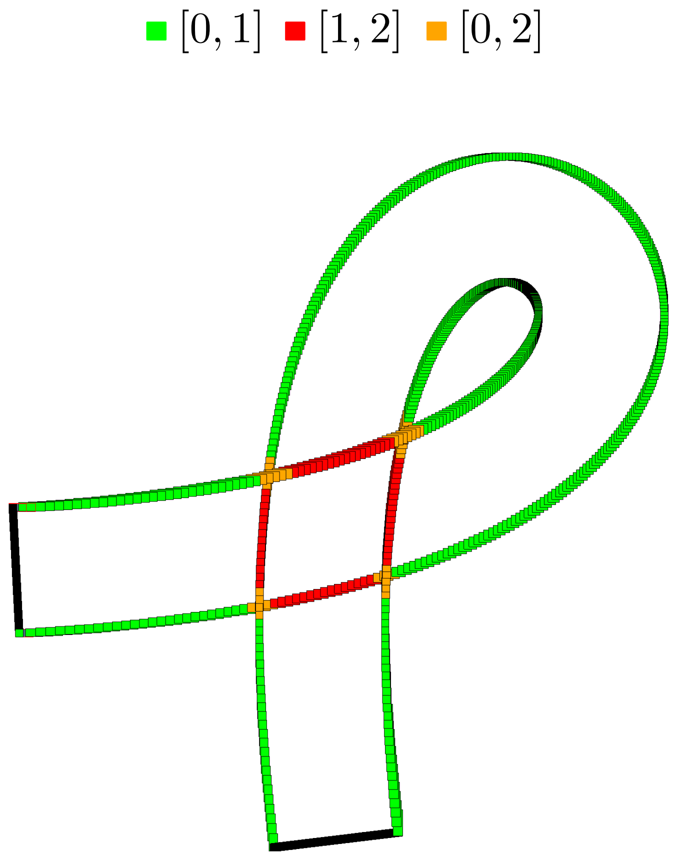


Results

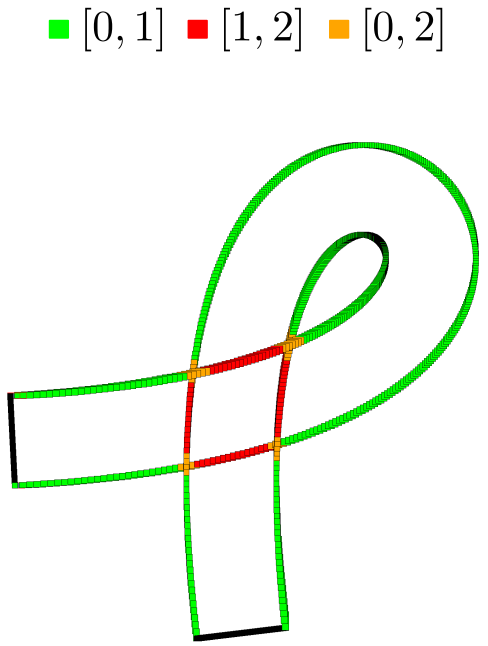
$$\mathbb{A}_{\mathbb{E}} = \{z \in \mathbb{R}^2 | \eta(\gamma(.), z) \neq 0\}$$



Results



Results



Detection of points of self intersection.

$$\eta^+ \geq \eta^- + 2$$

Conclusions

Using the topological properties of the robot's exteroceptive sensors contour, we are able to

- determine the are explored during a mission,
- determine the number of times a point in the space was in the robot's range of visibility.

Conclusions

Using the topological properties of the robot's exteroceptive sensors contour, we are able to

- determine the are explored during a mission,
- determine the number of times a point in the space was in the robot's range of visibility.

The explored area can be quickly computed using the property of continuity of winding numbers in the space and through the identification of self intersections on the contour.

Future Work

- Add uncertainty to the robot's trajectory and therefore to the explored area.
- Explore the property of self intersection of the current algorithm.

[1] A. Bethencourt and L. Jaulin.
Solving non-linear constraint satisfaction problems involving time-dependant functions.
Mathematics in Computer Science, 2014.

[2] S. G. Krantz.
The index or winding number of a curve about a point.
In *Handbook of Complex Variables*, pages 49–50, Boston, MA, 1999.

[3] Sliwka J. Jaulin L. Le Bars, F. and O. Reynet.
Set-membership state estimation with fleeting data.
Automatica, 2012.

[4] Dan Sunday.
Inclusion of a point in a polygon.
2001.