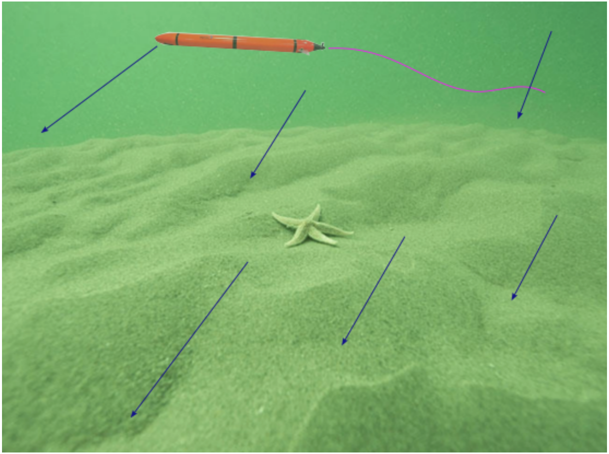
SLAM with magnetism only

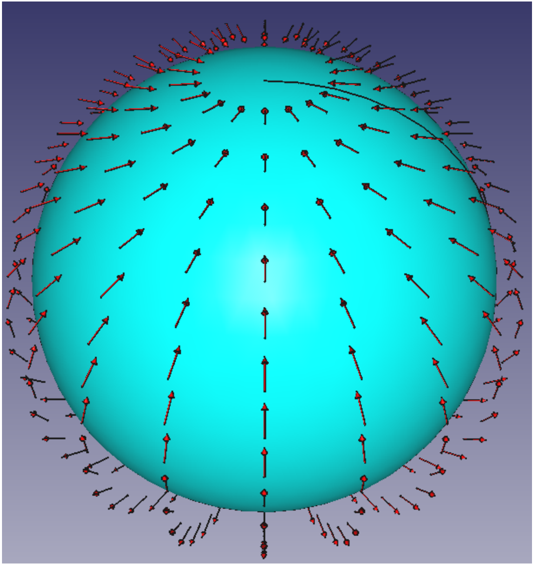
Luc Jaulin

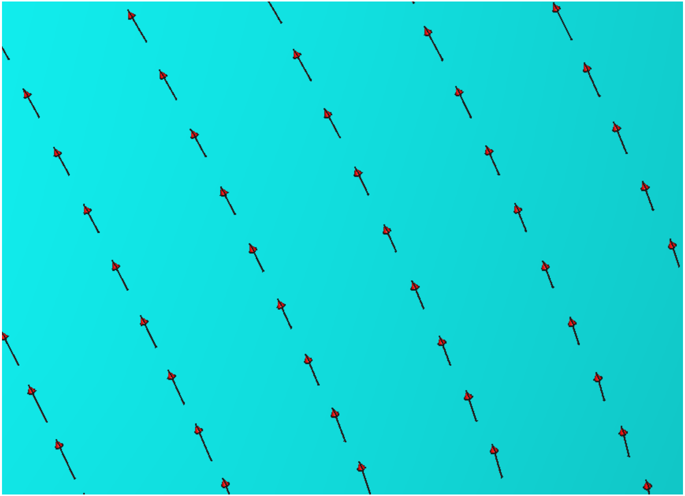
December 17, 2025, Robexday 5, Brest

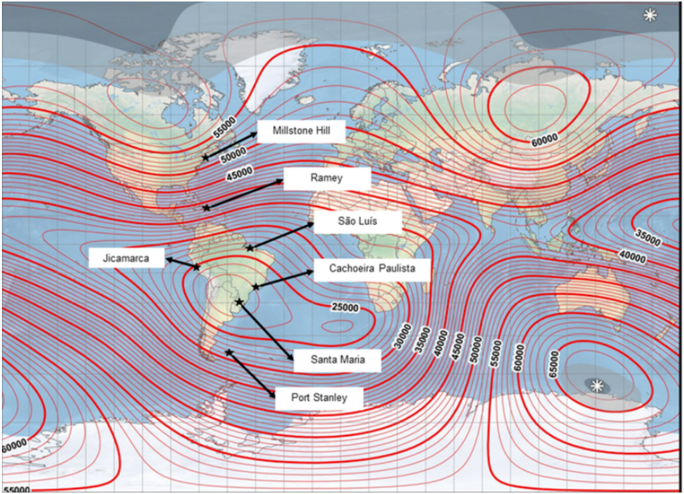


Explore and build a map in a magnetic environment

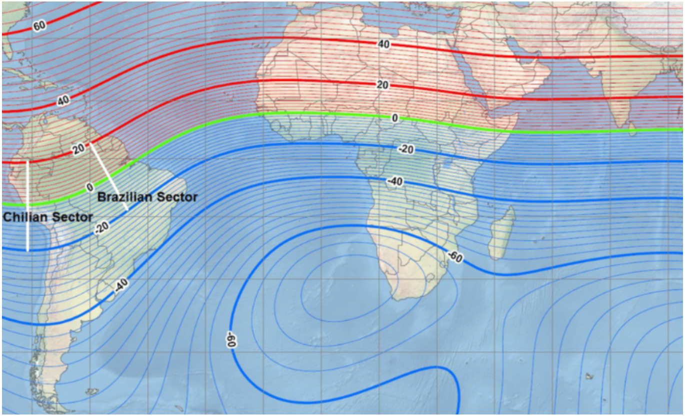
1. Magnetic world





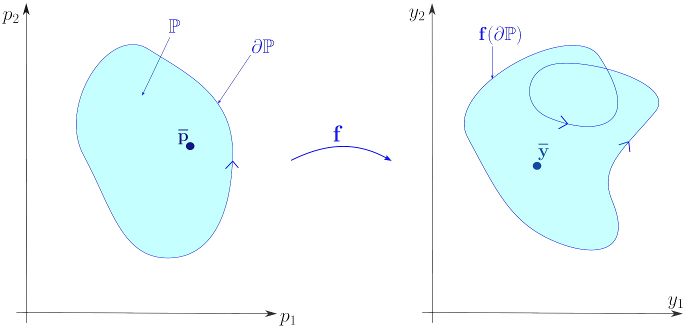


Intensity [nT]

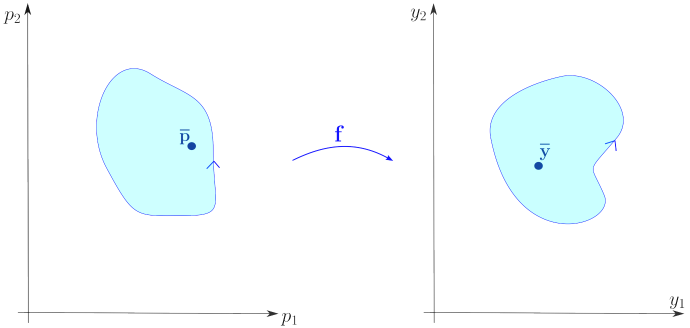


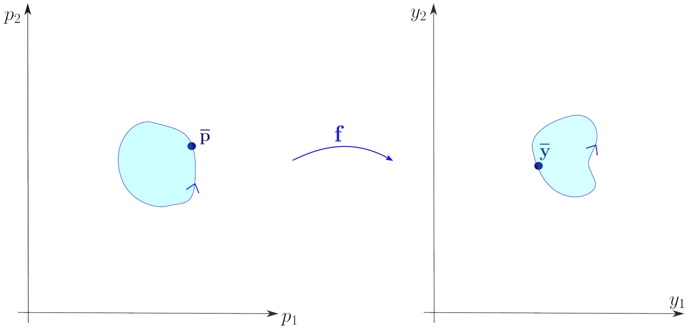
Inclinaison [deg]

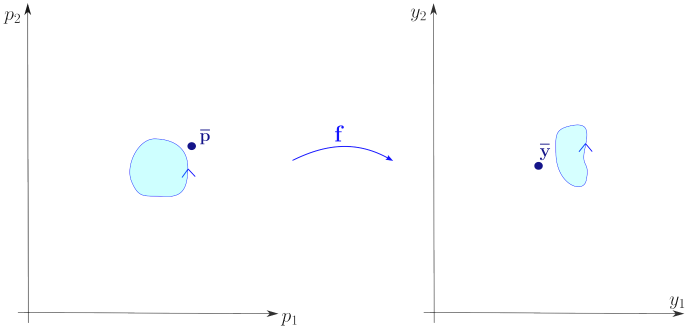
2. Topological degree



$$\deg(\mathbf{f}, \mathbb{P}, \bar{\mathbf{y}}) \neq 0 \Rightarrow \exists \bar{\mathbf{p}} \in \mathbb{P} \mid \mathbf{f}(\bar{\mathbf{p}}) = \bar{\mathbf{y}}$$


p_2
$\bar{\mathbf{p}}$
p_1
$\mathbf{f}$
y_2
$\bar{\mathbf{y}}$
y_1


p_2
$\overline{\mathbf{p}}$
p_1
$\mathbf{f}$
y_2
$\overline{\mathbf{y}}$
y_1


p_2
$\bar{\mathbf{p}}$
p_1
$\mathbf{f}$
y_2
$\bar{\mathbf{y}}$
y_1

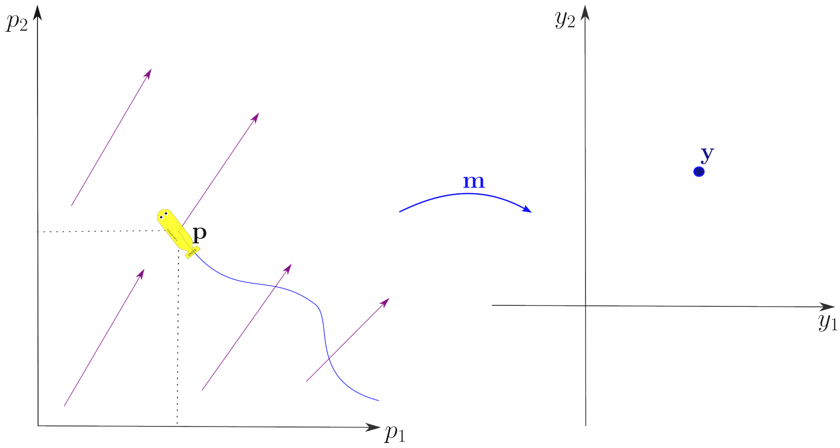
3. Mag-based navigation

$$\begin{cases} \dot{p}_1 & = & q_2 \cdot \cos q_1 \\ \dot{p}_2 & = & q_2 \cdot \sin q_1 \\ \dot{q}_1 & = & u_1 \\ \dot{q}_2 & = & u_2 \end{cases} \quad \text{and} \quad \begin{cases} y_1 & = & m_1(\mathbf{p}) & \qquad \text{(inclinaison)} \\ y_2 & = & m_2(\mathbf{p}) & \qquad \text{(intensity)} \end{cases}$$

Or more generally

$$\begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \end{pmatrix} = \mathbf{f}(\mathbf{p},\mathbf{q},\mathbf{u}) \text{ and } \left\{ \begin{array}{lcll} y_1 & = & m_1(\mathbf{p}) & \quad \text{(inclinaison)} \\ y_2 & = & m_2(\mathbf{p}) & \quad \text{(intensity)} \end{array} \right.$$

where $\mathbf{p} \in \mathbb{R}^2$ is the position.

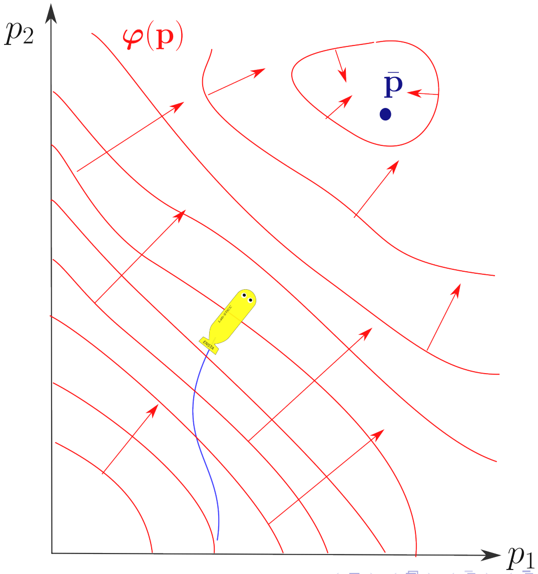

p_2
p_1
$\mathbf{p}$
$\mathbf{m}$
y_2
y_1
$\mathbf{y}$

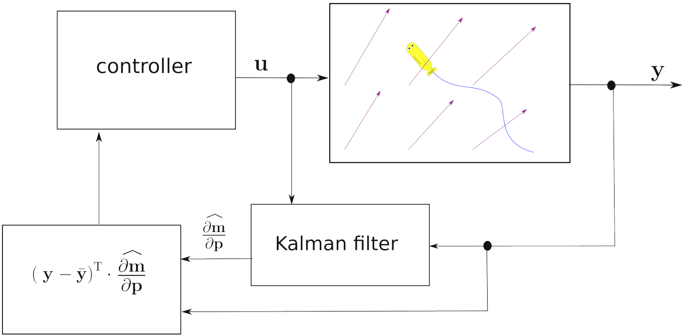
Go home

The robot wants to go home, *i.e.*, to a position $\bar{\mathbf{p}}$ where $\bar{\mathbf{y}} = \mathbf{m}(\bar{\mathbf{p}})$.
Think of the pigeon who returns to its loft using mag only.
It can move in order to minimize $\|\mathbf{m}(\mathbf{p}) - \bar{\mathbf{y}}\|^2$

Strategies: We build a clever field $\varphi(\mathbf{p})$ to follow.
For instance, for a *reach home*, we take

$$\begin{aligned} \varphi(\mathbf{p}) &= \frac{\partial}{\partial \mathbf{p}} \|\mathbf{m}(\mathbf{p}) - \bar{\mathbf{y}}\|^2 \\ &= 2\left(\mathbf{m}(\mathbf{p}) - \bar{\mathbf{y}}\right)^{\mathsf{T}} \cdot \frac{\partial \mathbf{m}}{\partial \mathbf{p}}(\mathbf{p}) \end{aligned}$$


p_2
$\varphi(\mathbf{p})$
$\bar{\mathbf{p}}$
p_1



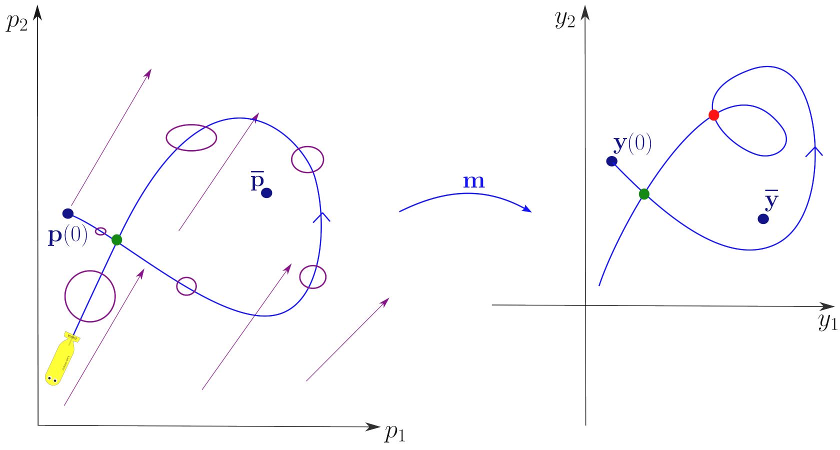
We need to estimate the variation $\frac{\partial \mathbf{m}}{\partial \mathbf{p}}$ using a Kalman filter

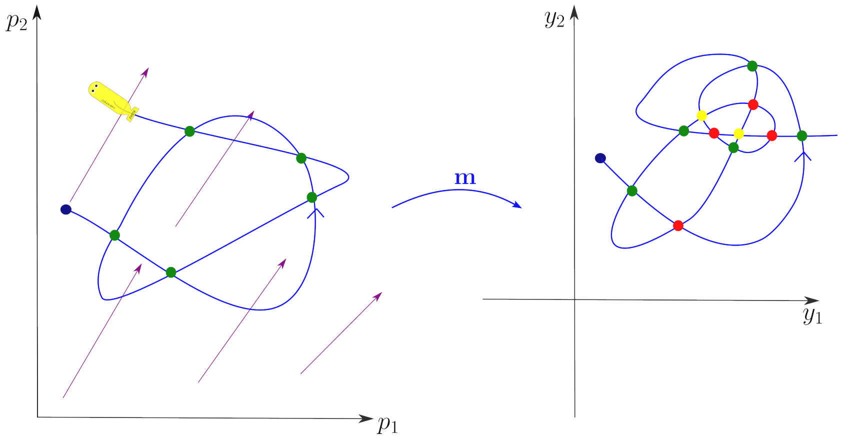
Unfortunately , we may converge to a local minimizer.

For SLAM, we need loops

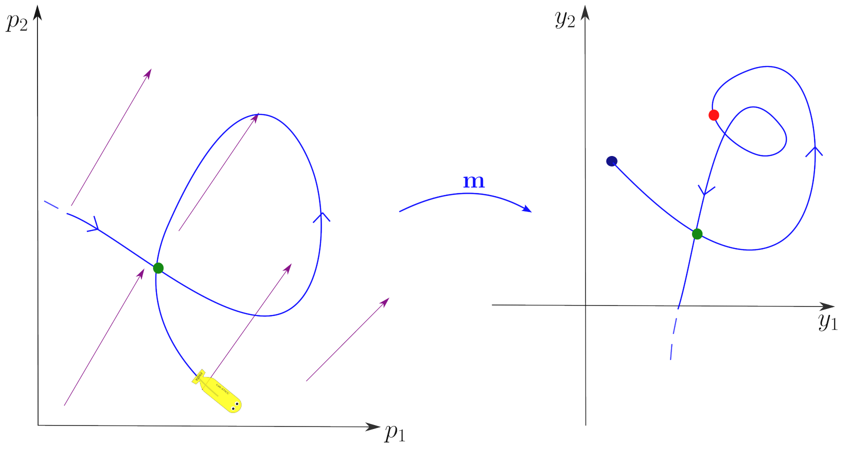
We are able

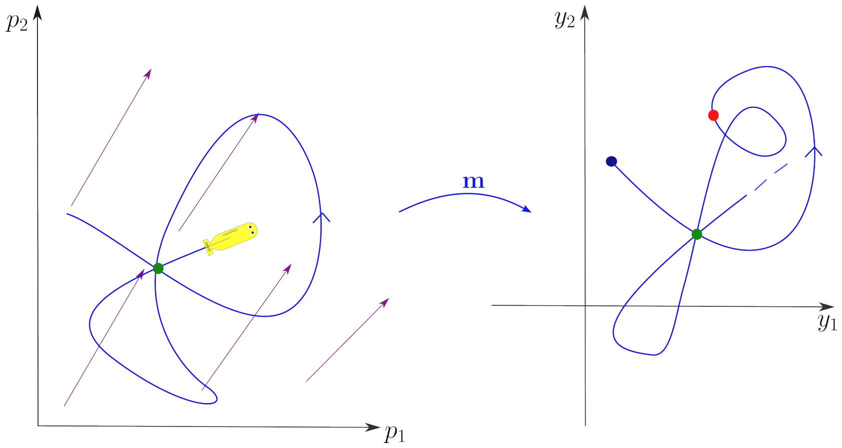
- to detect and prove the existence and uniqueness of a loop (topological degree)
- to prove the existence of $\mathbf{p}$ such that $\bar{\mathbf{y}} = \mathbf{m}(\mathbf{p})$.


p_2
p_1
$\bar{\mathbf{p}}$
$\mathbf{p}(0)$
$\mathbf{m}$
y_2
y_1
$\mathbf{y}(0)$
$\bar{\mathbf{y}}$

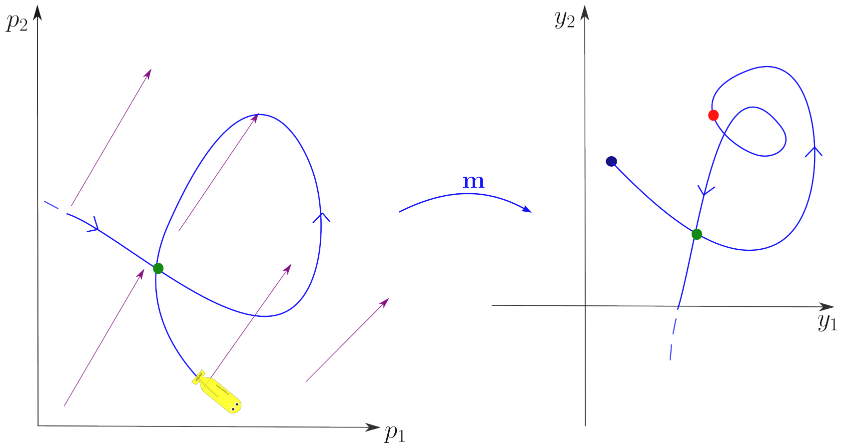

p_2
p_1
$\mathbf{m}$
y_2
y_1

Back to a past loop

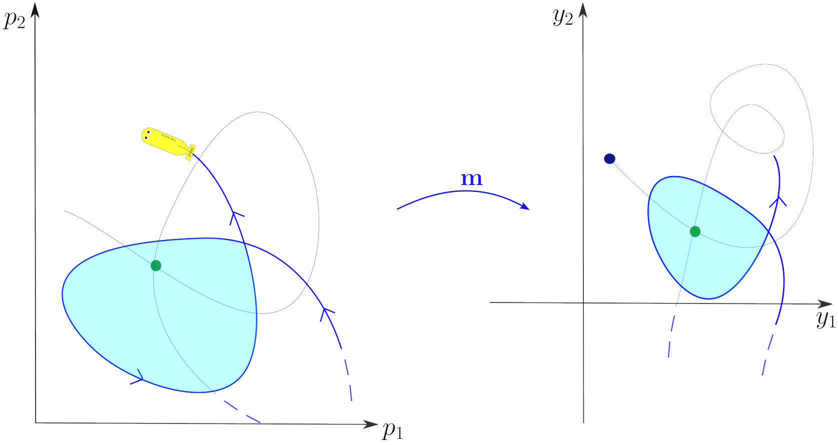

p_2
p_1
$\mathbf{m}$
y_2
y_1

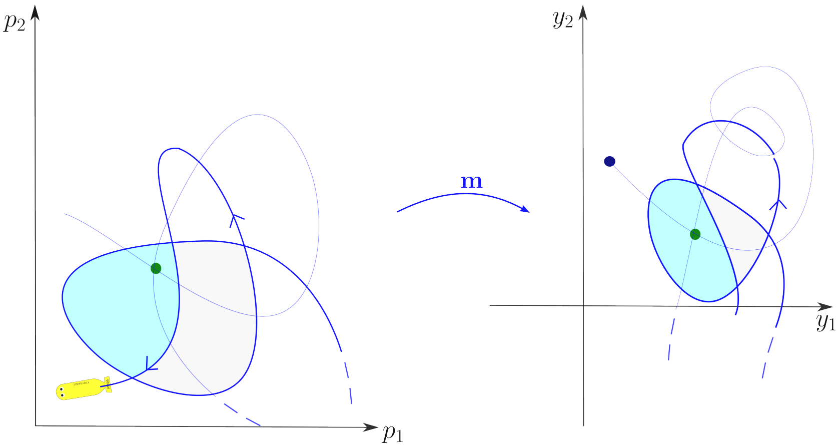


A three loop is impossible



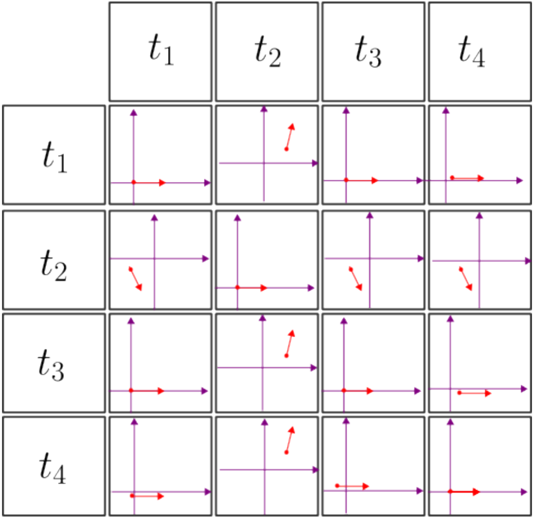
We need the topological degree.


p_2
p_1
$\mathbf{m}$
y_2
y_1


p_2
p_1
m
y_2
y_1

Toward a metric map: the pose matrix

We copy the representation by distance matrix.
The landmarks (nodes) are loop points.
They are associated to the north unit vector.
Nodes are linked by a pose (translation + rotation).



One true loop (t_1, t_3), two almost loops $(t_1, t_4), (t_3, t_4)$, No loop with t_2

References

1. Navigation with stable cycles [2]
2. Loops [4][1]
3. Brouwer for loop closure [3]

C. Aubry, R. Desmare, and L. Jaulin.
Loop detection of mobile robots using interval analysis.
Automatica, 2013.

Q. Brateau, F. L. Bars, and L. Jaulin.
Navigation without localization using stable cycles.
In *ICRA 2025, Submitted*, 2025.

M. Laranjeira, L. Jaulin, and S. Tauvry.
Underwater mosaics using navigation data and feature extraction.
Reliable Computing, 22:116–137, 2016.

S. Rohou, L. Jaulin, L. Mihaylova, F. L. Bars, and S. Veres.
Reliable Robot Localization.
Wiley, dec 2019.