Guaranteed Polynesian Navigation

Thibaut Nico[1], Luc Jaulin *[2], and Benoît Zerr[2]

[1]ECA, Lab-STICC UMR CNR 6285
[2]ENSTA-Bretagne, Lab-STICC UMR CNRS 6285

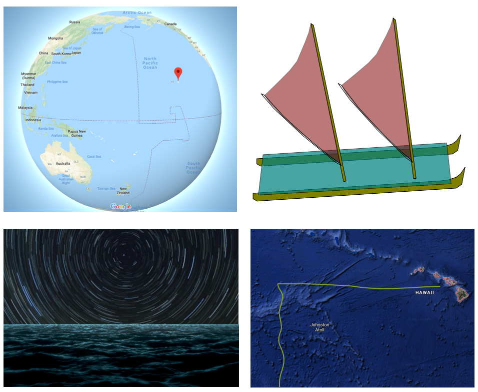

Figure 1: Polynesian navifation

Keywords: Navigation; Intervals, Contractors, No-lost zone

Introduction

The Polynesian navigation problem asks to move from islands to other islands without being lost. The navigation should be performed without without GPS, compass and clocks. The difficulty of the navigation is illustrated by Figure 1: the ocean is huge, the islands are small, the boats are more or less uncertain.

Among the techniques used by Polynesian, the observation of the stars (see Figure 2) are useful to get the heading, but also to detect if the boat is on the route which leads us to the desired island. The approach we will follow to guarantee that we can reach an island from another island, uses guaranteed integration [10], tube programming [9], [8], constraint programming [11], localization [3], contractors [2] and interval analysis [6][4].

*Corresponding author.

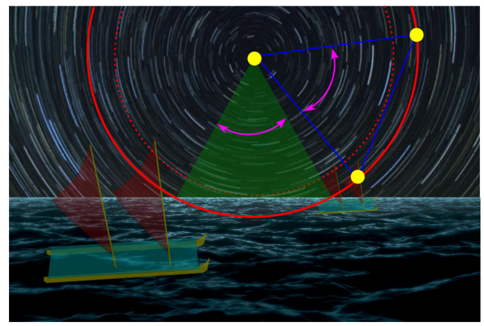

Figure 2: Pair of stars technique: the boat is on the right route if the bottom star rises when the right star sets

Formalisation

The problem can be formalized as follows

- Given a set of geo-localized islands $\mathbf{m}_i, i \geq 0$.
- The ith coastal area is:

$$\mathbb{C}_i = \{\mathbf{x} \,|\, c_i(\mathbf{x}) \leq 0\}.$$

- A robot has to move in this environment without being lost.

Figure 3 represents a set of 4 islands with the associated coastal zones $\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3, \mathbb{C}_4$ (painted blue).

We assume the following

- The coastal areas are small compare to the offshore area.
- In the coastal area, the robot knows its state.
- Offshore, the robot is blind and has an open loop strategy, such as for instance go North.

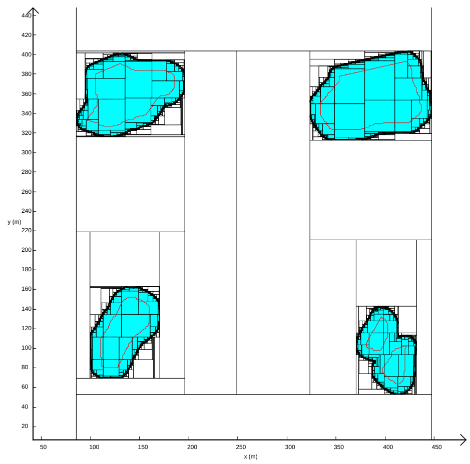

Figure 3: Islands ans coastal zones

- The robot is described by blind state equations

$$\left\{ \begin{array}{l} \dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right), \quad \mathbf{u}\left(\cdot\right) \in \left[\mathbf{u}\right]\left(t\right) \\ \mathbf{x}\left(0\right) = \mathbf{x}_0 \end{array} \right.$$

where the input $\mathbf{u}(t)$ belongs to the uncertainty box $\left[\mathbf{u}\right]\left(t\right)$.

We define the *set flow* $\mathbf{\Phi} : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathcal{P}\left(\mathbb{R}\right)$ as:

$$\begin{array}{rl} \mathbf{\Phi}(t_1, \mathbf{x}_0) & = \{ \quad \mathbf{a} | \exists \mathbf{u}\left(\cdot\right) \in \left[\mathbf{u}\right]\left(t\right), \mathbf{a} = \mathbf{x}(t_1), \\ & \qquad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \mathbf{x}\left(0\right) = \mathbf{x}_0 \qquad \} \end{array}$$

Given the set $\mathbb{A}$ (for instance a coastal area), the *backward reach set* [1] is defined by

$$\begin{array}{rl} \mathrm{Back}(\mathbb{A}) = \{ & \mathbf{x} \mid \forall \boldsymbol{\varphi} \in \mathbf{\Phi}, \\ & \exists t \geq 0, \boldsymbol{\varphi}(t, \mathbf{x}) \in \mathbb{A} \quad \} \end{array}$$

Interval analysis is often used to compute backward reach sets in the case where the robot is nonlinear [5], [7]. We have

$$\mathrm{Back}(\mathbb{A} \cup \mathbb{B}) \quad \supset \quad \mathrm{Back}(\mathbb{A}) \cup \mathrm{Back}(\mathbb{B}) \ .$$

This is the *Archipelago effect* which tells us that finding an Archipelago $(\mathbb{A} \cup \mathbb{B})$ is easier than finding individual islands, as illustrated by Figures 4 and 5.

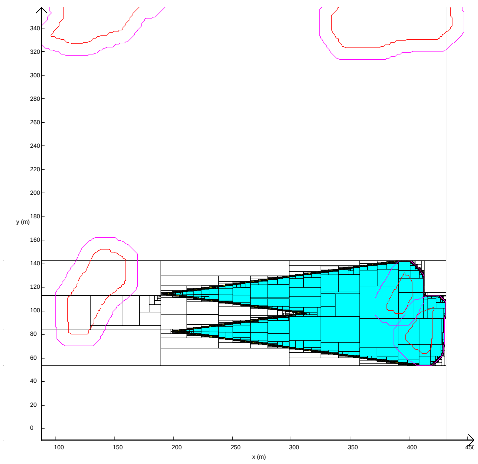

Figure 4: $\mathrm{Back}(\mathbb{A}) \cup \mathrm{Back}(\mathbb{B})$

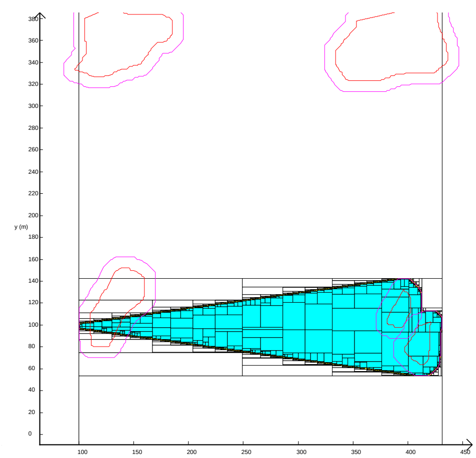

Figure 5: $\mathrm{Back}(\mathbb{A} \cup \mathbb{B})$

Moving between coastal zones

Assume that we have m coastal sets $\mathbb{C}_1, \mathbb{C}_2, \dots, i \in \{1, 2, \dots\}$ and open loop control strategies $\mathbf{u}_j, j \in \{1, 2, \dots\}$ or equivalently, we have set flows $\mathbf{\Phi}_j(t, \mathbf{x}_0)$. Moreover, we assume that the control strategy cannot change offshore. As a consequence,

- From $\mathbb{C}_1$ we can reach $\mathbb{C}_2$ with the jth control strategy if $\mathbb{C}_1 \cap \mathrm{Back}(j, \mathbb{C}_2) \neq \emptyset$.

- From $\mathbb{C}_1$ we can reach $\mathbb{C}_2$ with at least one control strategy if $\mathbb{C}_1 \cap \bigcup_j \mathrm{Back}(j, \mathbb{C}_2) \neq \emptyset$.

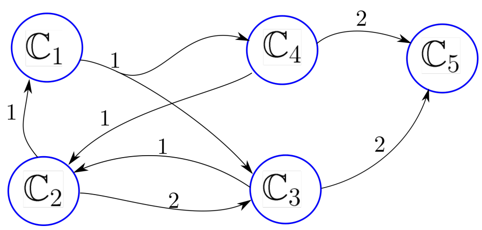

Figure 6: Reach an island from another island using a 'Go-East' strategy

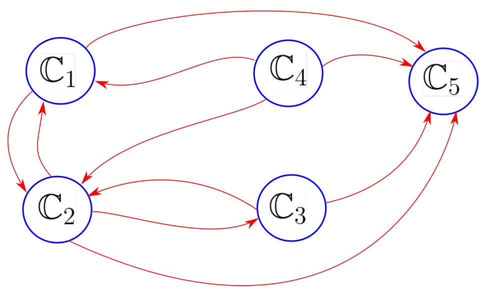

Figure 7: Reachability graph

- From $\mathbb{C}_1$ we can reach $\mathbb{C}_2 \cup \mathbb{C}_3$ with at least one control strategy if $\mathbb{C}_1 \cap \bigcup_j \mathrm{Back}(j, \mathbb{C}_2 \cup \mathbb{C}_3) \neq \emptyset$.

Therefore, we define the *reachability* relation $\hookrightarrow$ as:

- $\mathbb{C}_a \hookrightarrow \mathbb{C}_b$ if from $\mathbb{C}_a$ we can reach $\mathbb{C}_b$ with at least one control strategy j.
- $\hookrightarrow$ is the smallest transitive relation which satisfies

$$\left\{ \begin{array}{c} \forall k \in \mathbb{K}, \mathbb{C}_{i_k} \hookrightarrow \mathbb{C}_b \\ \exists j, \mathbb{C}_a \cap \mathrm{Back}(j, \bigcup_{k \in \mathbb{K}} \mathbb{C}_{i_k}) \neq \emptyset \end{array} \right. \Rightarrow \mathbb{C}_a \hookrightarrow \mathbb{C}_b$$

Consider for instance, the hyper-graph of Figure 6 where the relation $\mathbb{A} \xrightarrow{j} \mathbb{B}, \mathbb{C}$ means that from $\mathbb{A}$ the robot can reach either $\mathbb{B}$ or $\mathbb{C}$ using the jth strategy. For instance, in our graph

$$\mathbb{C}_1 \cap \mathrm{Back}(1, \mathbb{C}_3 \cup \mathbb{C}_4) \neq \emptyset \Rightarrow \mathbb{C}_1 \xrightarrow{1} (\mathbb{C}_3, \mathbb{C}_4)$$

Thus, the associated reachability graph (corresponding to $\hookrightarrow$) is given by Figure 7.

In a similar way, we can also define the forward reach set as illustrated by Figure 8.

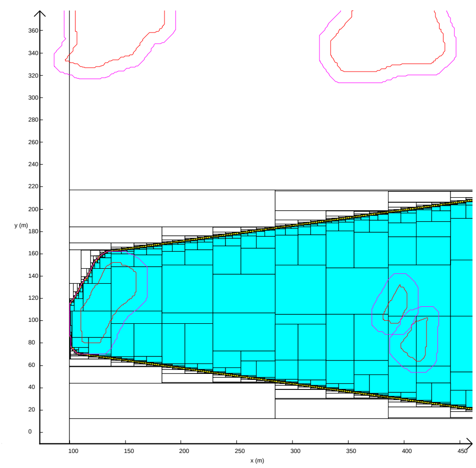

Figure 8: Forward reach set from a given island

No lost zone

We define the no-lost zone as the set $\mathbb{S}$ of all states that we may visit from a coastal area without being lost with the available control strategies. Define the *index set* associated with the strategy $\mathcal{I}_j$ as

$$\mathcal{I}_j = \{k | \mathbb{C}_k \cap \mathrm{Back}(j, \bigcup_{i \neq k} \mathbb{C}_i) \neq \emptyset\}.$$

If we start from $\mathbb{C}_k$,$k \in \mathcal{I}_j$, then we will reach at least another coastal area with the control strategy j. We have

$$\left\{ \begin{array}{c} \mathbf{x} \in \mathrm{Back}(j, \bigcup_i \mathbb{C}_i) \\ \mathbf{x} \in \mathrm{Forw}(j, \mathbb{C}_k), k \in \mathcal{I}_j \end{array} \right. \Rightarrow \mathbf{x} \in \mathbb{S}$$

Thus

$$\mathbb{S} \subset \bigcup_j \bigcup_{k \in \mathcal{I}_j} \mathrm{Forw}(j, \mathbb{C}_k) \cap \mathrm{Back}(j, \bigcup_i \mathbb{C}_i).$$

This property will allow us to have an inner approximation of the no-lost zone, which is the main contribution of this paper. This is illustrated by Figure 9 with 8 strategies: North, East, South, West, North-East, East-South, South-West, West-North.

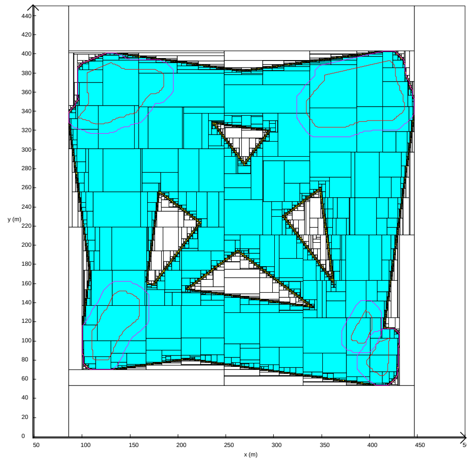

Figure 9: No-lost zone associated with the 5 islands

Aknowledgement

This work has been supported by the French Government Defense procurement and technology agency (DGA).It also benefited from the support of the project CONTREDO of the French National Research Agency (ANR).

References

[1] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Springer Science & Business Media, Oct. 2007.

[2] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173:1079–1100, 2009.

[3] V. Drevelle, L. Jaulin, and B. Zerr. Guaranteed characterization of the explored space of a mobile robot by using subpavings. In *Proc. Symp. Nonlinear Control Systems (NOLCOS'13)*, Toulouse, 2013.

[4] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl. Computational complexity and feasibility of data processing and interval computations. *Reliable Computing*, 4(4):405–409, 1997.

[5] T. L. Mézo, L. Jaulin, and B. Zerr. An interval approach to compute invariant sets. *IEEE Transaction on Automatic Control*, 62:4236–4243, 2017.

[6] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.

[7] N. Ramdani and N. Nedialkov. Computing Reachable Sets for Uncertain Nonlinear Hybrid Systems using Interval Constraint Propagation Techniques. *Nonlinear Analysis: Hybrid Systems*, 5(2):149–162, 2011.

[8] S. Rohou. *Reliable robot localization: a constraint programming approach over dynamical systems*. PhD dissertation, Université de Bretagne Occidentale, ENSTA-Bretagne, France, december 2017.

[9] S. Rohou, L. Jaulin, M. Mihaylova, F. L. Bars, and S. Veres. Guaranteed Computation of Robots Trajectories. *Robotics and Autonomous Systems*, 93:76–84, 2017.

[10] J. A. D. Sandretto and A. Chapoutot. Validated simulation of differential algebraic equations with runge-kutta methods. *Reliable Computing*, 22, 2016.

[11] G. Trombettoni and G. Chabert. Constructive Interval Disjunction. In *Proc. CP, Constraint Programming*, pages 635–650, LNCS 4741, 2007.