Secure the zone from intruders with a group robots

Khadimoullah Vencatasamy, Luc Jaulin and Benoît Zerr

Lab-STICC, ENSTA Bretagne, Brest, France

lucjaulin@gmail.com

Abstract. The paper proposes new method to secure a zone of the world from intruders using a group a robots. The principle is to control the group in order to form a chain through which no intruder can go without being detected. We use a set membership method based on interval analysis and mathematical morphology to guarantee that no intruder is inside the secure zone. The approach is illustrated by an example where the environment is the Bay of Biscay, the intruder is a submarine and the group of robots consists of small underwater autonomous vehicles.

1 Introduction

We consider n robots $\mathcal{R}_1, \ldots, \mathcal{R}_n$ at positions $\mathbf{a}_1, \ldots, \mathbf{a}_n$ and moving in a 2D world [1]. Each robot has a visibility zone. If an intruder is inside the visibility zone of one robot, it is detected. The robots have to collaborate to guarantee that there is no moving intruder inside a subzone of a compact subset $\mathbb{O}$ of $\mathbb{R}^2$, representing the world.

This project is motivated by the detection of submarine intruders inside the Bay of Biscay (golfe de Gascogne) by low cost robots [2] equipped with low cost and passive sensors (microphones, cameras [3], electric sense [4][5]). The set $\mathbb{O}$ corresponds to the ocean and the intruder is a submarine considered as an enemy. We will avoid using expensive sonars such as side-scan sonars [6] which consume energy and which are not discrete.

For simplicity, we consider a 2D environment, and the robots are able find their position with a given accuracy using a state observer. We assume that each robot is able to detect any intruder inside a disk of a known radius. Moreover, the intruder's dynamics is assumed to be known (but not its control).

The goal of this paper is twofold: (1) to characterize a set for which we can guarantee that there is no intruder (this corresponds to the secure zone) and (2) to find a control strategy for the group of robots in order to extend the secure zone as much as possible. For this purpose, we will use interval analysis and control theory.

Set-membership approaches have mainly been used in the context of underwater robotics [7][8][9] for guaranteed localization and mapping. For the first time, we use it here to guarantee that a zone is secured.

In this paper, we are not interested in a dynamical control of one robot in order to perform a particular mission, but more by a collective control of a group of robots for a common task.

The paper is organized as follows. Section 2 recalls how to control independently a group of robots that cannot communicate. Section 3 shows how to compute an inner and outer approximation of the secure zone. An illustration of our approach is given in Section 4 where a group of robots collaborate to secure the Bay of Biscay. Section 5 concludes this paper and proposes some perspectives.

2 Control of a group of robots

The first step to be solved to secure a zone is to control a group of robots assuming that the robots cannot communicate, in order to be as silent as possible. In such a case, classical linear control [10] do not apply. The general method in this situation is the flocking approach [11] or to consider that all trajectories of all robots have been decided a priori.

Remark. When we say that the robots do not communicate, it means that they do not share any information stored inside their memory. But, we can still assume that each robot, via its own sensors are able estimate the state (position, orientation and speed) of its neighbors. We do not consider the perception of neighbors as a communication. Now, since the behavior of each robot is influenced by what is stored in its memory, a robot can guess some part of the memory of its neighbors by analyzing their behaviors. Moreover, the robots can voluntarily change their behaviors in order to transmit a coded message. It is what insects do when they perform some kind of dance to communicate. For instance, by performing the figure-eight dance, the honey bees can share with other members of the colony, information about the direction and distance to flowers yielding nectar and pollen, to water sources, or to new nest-site location. In such a case, there is a clear intention to communicate. Here, we shall consider that we have no communication when there is no intention to transmit any message from one robot to another robot.

2.1 Flocking

To illustrate the flocking, let us consider m robots described by the following state equations:

$$\left\{ \begin{array}{rcl} \dot{x}_i & = & \cos\theta_i \\ \dot{y}_i & = & \sin\theta_i \\ \dot{\theta}_i & = & u_i \end{array} \right.$$

The state vector is $\mathbf{x}(i) = (x_i, y_i, \theta_i)$. In this model, the speed in always equal to 1, which means that each robot is only control its heading through the input u_i. These robots can see all other robots, but are not able to communicate with them. We want that these robots behave as a flock as illustrated by Figure 1. Basic models of flocking behaviors are controlled by three rules of Reynolds:

- The *separation* corresponds to a short range repulsion force. This allows the flock to avoid crowding neighbors.
- The *alignment* force implies that each robot steers towards average heading of its neighbors.
- The *cohesion* is a long range attraction where each robot of the flock is attracted by its neighbors.

Using a artificial potential field method [12] (see also [13] or [14] in the context of ocean robotics) , we can find a controller for each robot to obtain a flock. In Figure 1, the robots are first initialized randomly in the same zone of the world. Then, following the rules of Reynolds, they adopt the behavior of a swarm.

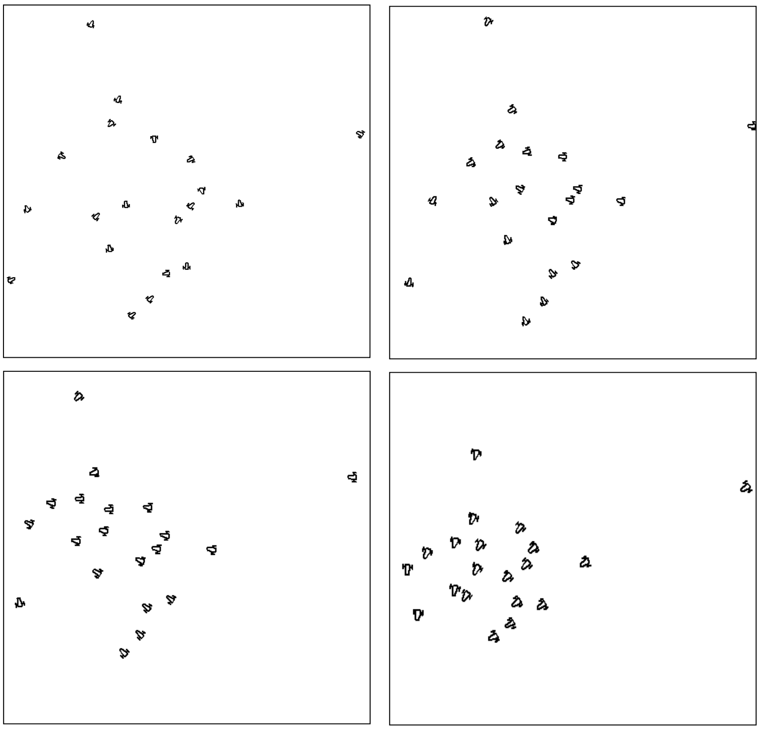

Figure 1: Illustration of the flocking behavior from a random initialization in the case where $m = 20$

For this, we define the potential of the ith robot as

$$V_i = \sum_{j \neq i} \alpha \left\|\mathbf{p}(i) - \mathbf{p}(j)\right\|^2 + \frac{\beta}{\left\|\mathbf{p}(i) - \mathbf{p}(j)\right\|}$$

where $\mathbf{p}(i) = (x_i, y_j)$. The quadratic part of the sum corresponds to the cohesion and the hyperbolic term corresponds to the separation. The coefficients α and β have to be tuned. For instance, if β is high, the repulsion will be stronger and the flock will be more scattered. The gradient of V_i is

$$\frac{dV_i}{d\mathbf{p}(i)} = \sum_{j \neq i} \left(2\alpha \left(\mathbf{p}(i) - \mathbf{p}(j)\right) - \frac{\beta \left(\mathbf{p}(i) - \mathbf{p}(j)\right)^T}{\left\|\mathbf{p}(i) - \mathbf{p}(j)\right\|^3} \right).$$

Since we want to decrease V_i, we have to follow the opposite direction of the gradient. Moreover, in order to follow the alignment rule of Reynolds each robot should get the same heading as the other robots. Thus, the controller is built in order to follow the direction

$$\sum_{j \neq i} \left(-2\alpha \left(\mathbf{p}(i) - \mathbf{p}(j)\right) - \frac{\beta \left(\mathbf{p}(i) - \mathbf{p}(j)\right)^T}{\left\|\mathbf{p}(i) - \mathbf{p}(j)\right\|^3} + \begin{pmatrix} \cos\theta_j \\ \sin\theta_j \end{pmatrix} \right).$$

The flocking is a nice method to move a group of robots and could be used to find an intruder. Moreover, no communication or distributed computation is needed. Now we cannot guarantee anything with flocking behavior about the stability or the cohesion (see e.g., [1]). We need a control which is more deterministic as proposed in the following subsection.

2.2 Group of robots forming a chain

To secure a zone, with guarantee, the first idea that comes to mind is to build a virtual chain made with robots. If the chain is built with the right spacing and takes into account the visibility zone of each robot, the intruder will not be able to cross the chain without being detected. Moreover, we want the robots move with respect to a cycle so that they can be able to recharge their batteries at some points of the cycle.

To illustrate how this can be done, assume that we have a group of $m = 20$ robots the motion of which is described by the state equation

$$\left\{ \begin{array}{lll} \dot{x}_1 & = & x_4 \cos x_3 \\ \dot{x}_2 & = & x_4 \sin x_3 \\ \dot{x}_3 & = & u_1 \\ \dot{x}_4 & = & u_2 \end{array} \right.$$

where (x_1, x_2) corresponds to the position of the robot, x_3 to its heading and x_4 to its speed. We want the robots to follow an ellipse which corresponds to a barrier for the intruder.

Circle. Assume first that this ellipse is a circle. We want a controller for each of these robots so that the ith robot follows the trajectory

$$\left(\begin{array}{c} \cos(at + \frac{2i\pi}{m}) \\ \sin(at + \frac{2i\pi}{m}) \end{array} \right),$$

where $a = 0.1$. As a consequence, after the initialization step, all robots are uniformly distributed on the unit circle, turning around the origin. To do this, we can use a feedback linearization method taking as an output $\mathbf{y} = (x_1, x_2)$. We get the controller

$$\mathbf{u} = \left(\begin{array}{cc} -x_4 \sin x_3 & \cos x_3 \\ x_4 \cos x_3 & \sin x_3 \end{array} \right)^{-1} \cdot \left(\mathbf{c}\,(t) - \left(\begin{array}{c} x_1 \\ x_2 \end{array} \right) + 2\dot{\mathbf{c}}\,(t) - 2 \left(\begin{array}{c} x_4 \cdot \cos(x_3) \\ x_4 \cdot \sin(x_3) \end{array} \right) + \ddot{\mathbf{c}}\,(t) \right), \quad (1)$$

where $\mathbf{c}(t)$ is the desired position. For the ith robot, we take

$$\mathbf{c}\,(t) = \left(\begin{array}{c} \cos(at + \frac{2i\pi}{m}) \\ \sin(at + \frac{2i\pi}{m}) \end{array} \right), \dot{\mathbf{c}}\,(t) = a \cdot \left(\begin{array}{c} -\sin(at + \frac{2i\pi}{m}) \\ \cos(at + \frac{2i\pi}{m}) \end{array} \right), \ddot{\mathbf{c}}\,(t) = -a^2 \mathbf{c}\,(t) \cdot$$

Ellipse. By using a linear transformation of the unit circle, we can change the controllers for the robots so that all robots stay on a moving ellipse. For instance, assume that we want a first axis of length $20 + 15 \cdot \sin(at)$ and the second axis of length 20. Moreover, we want the ellipse rotating by choosing an angle for the first axis of $\theta = at$.

To get the right ellipse, for each $\mathbf{c}(t)$, we apply the transformation

$$\mathbf{w}(t) = \underbrace{\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}}_{=\mathbf{R}} \cdot \underbrace{\begin{pmatrix} 20 + 15 \cdot \sin(at) & 0 \\ 0 & 20 \end{pmatrix}}_{=\mathbf{D}} \cdot \mathbf{c}(t)$$

where $\mathbf{w}(t)$ is the new desired position. To apply our controller, we also need the two first derivatives of $\mathbf{w}(t)$. We have

$$\dot{\mathbf{w}} = \mathbf{R} \cdot \mathbf{D} \cdot \dot{\mathbf{c}} + \mathbf{R} \cdot \dot{\mathbf{D}} \cdot \mathbf{c} + \dot{\mathbf{R}} \cdot \mathbf{D} \cdot \mathbf{c},$$

where

$$\dot{\mathbf{D}} = \begin{pmatrix} 15a \cdot \cos(at) & 0 \\ 0 & 0 \end{pmatrix}, \text{ and } \dot{\mathbf{R}} = a \cdot \begin{pmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix}.$$

Moreover

$$\ddot{\mathbf{w}} = \mathbf{R} \cdot \mathbf{D} \cdot \ddot{\mathbf{c}} + \mathbf{R} \cdot \ddot{\mathbf{D}} \cdot \mathbf{c} + \ddot{\mathbf{R}} \cdot \mathbf{D} \cdot \mathbf{c} + 2 \cdot \dot{\mathbf{R}} \cdot \mathbf{D} \cdot \dot{\mathbf{c}} + 2 \cdot \mathbf{R} \cdot \dot{\mathbf{D}} \cdot \dot{\mathbf{c}} + 2 \cdot \dot{\mathbf{R}} \cdot \dot{\mathbf{D}} \cdot \mathbf{c}$$

where

$$\ddot{\mathbf{D}} = \begin{pmatrix} -15a^2 \cdot \sin(at) & 0 \\ 0 & 0 \end{pmatrix} \text{ and } \ddot{\mathbf{R}} = -a^2 \cdot \mathbf{R}.$$

We can now apply the controller given by Equation (1), where $\mathbf{c}(t)$ is now replaced by $\mathbf{w}(t)$. Figure 2 illustrates the behavior of the control law and shows that the group moves exactly with the ellipse.

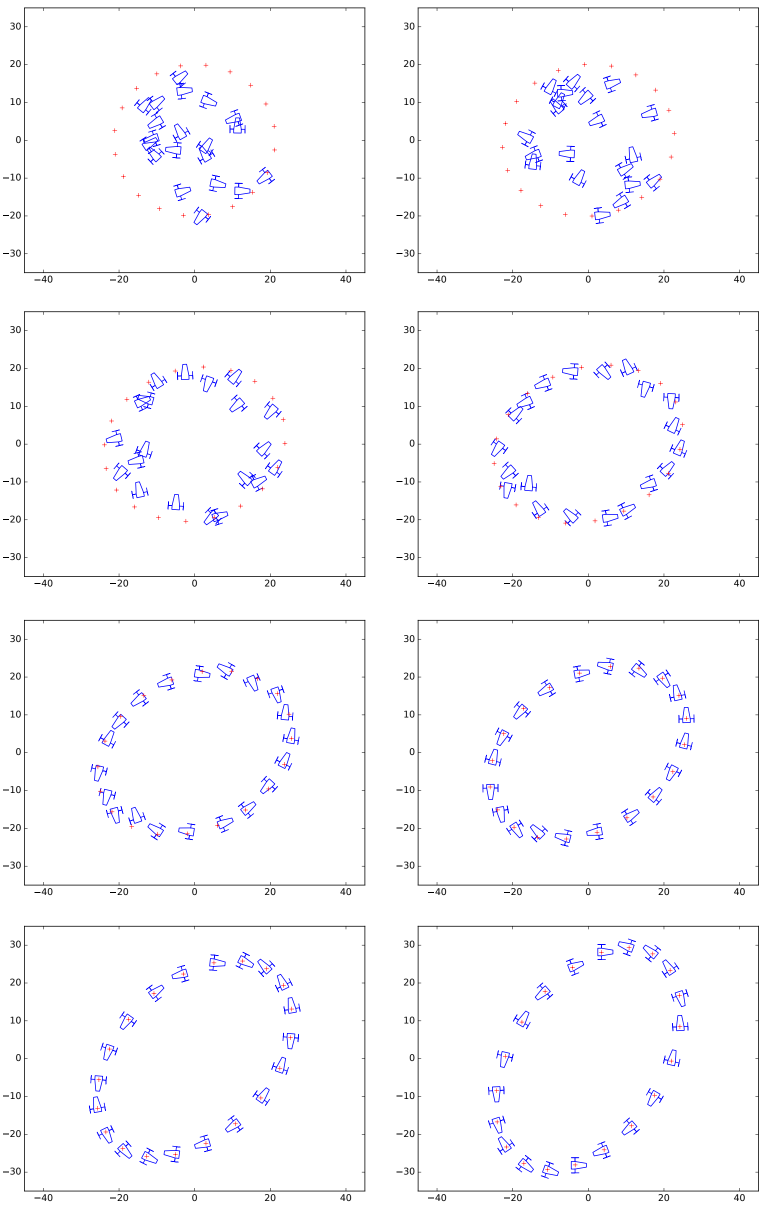

Figure 2: During the transient phase, all robots go toward their setpoints in the ellipse. Then, the group follows exactly the deformation of the ellipse.

We have shown how a group of robots can follow an ellipse. As a result, we may interpret the group as a unique *ellipse robot* which can have an arbitrary shape. Now, when the ellipse becomes flat (as it is the case when we want to build a chain for the intruders), the placement of the robots is not uniform: the robots concentrate around the two apogee points. The next subsection explains how to get a uniform distribution of robots along the ellipse.

2.3 Convoy

To illustrate how to have a uniform distribution along the ellipse (with a constant distance between robots), consider one robot $\mathcal{R}_A$ described by the following state equations:

$$\begin{cases} \dot{x}_a &= v_a \cos \theta_a \\ \dot{y}_a &= v_a \sin \theta_a \\ \dot{\theta}_a &= u_{a1} \\ \dot{v}_a &= u_{a2} \end{cases}$$

where v_a is the speed of $\mathcal{R}_A$ the robot, θ_a its orientation and (x_a, y_a) the coordinates of its center. The robot can be considered as a leader and all robots have to follow it, maintaining a constant distance between two successors. For instance, as previously, we can ask all robots to follow $\mathcal{R}_A$. Therefore, we ask $\mathcal{R}_A$ to follow the trajectory:

$$\begin{cases} \hat{x}_a(t) &= L_x \sin(\omega t) \\ \hat{y}_a(t) &= L_y \cos(\omega t) \end{cases}$$

with $\omega = 0.1$, $L_x = 20$ and $L_y = 5$. We can take:

$$\mathbf{u}_a = \begin{pmatrix} -v_a \sin \theta_a & \cos \theta_a \\ v_a \cos \theta_a & \sin \theta_a \end{pmatrix}^{-1} \cdot \left(\begin{pmatrix} \hat{x}_a \\ \hat{y}_a \end{pmatrix} - \begin{pmatrix} x_a \\ y_a \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} \hat{x}_a \\ \hat{y}_a \end{pmatrix} - \begin{pmatrix} v_a \cos \theta_a \\ v_a \sin \theta_a \end{pmatrix} \right)$$

where

$$\begin{array}{rclcrcl} \hat{x}_a &=& L_x \sin \omega t &,& \frac{d\hat{x}_a}{dt} &=& \omega L_x \cos \omega t \\ \hat{y}_a &=& L_y \cos \omega t &,& \frac{d\hat{y}_a}{dt} &=-& \omega L_y \sin \omega t. \end{array}$$

We now want that $m = 6$ other robots with the same state equations follow this robot taking exactly the same path. The distance between two robots should be $d = 5m$. To achieve this goal, we propose to save every $ds = 0.1m$ the value of the state of $\mathcal{R}_A$ and to communicate this information to the m followers. This will allow us to synchronize the time with the traveled distance. For this, we propose to add a new state variable s to $\mathcal{R}_A$ which corresponds to the curvilinear value that could have been measured by a virtual odometer. Each time the distance ds has been measured by the virtual odometer, s is initialized to zero and the value for the state of $\mathcal{R}_A$ is broadcast. We have the new state equation $\dot{s} = v$ which corresponds to the virtual odometer. Each time s is larger than $ds = 0.1m$, we set $s = 0$ and we store the corresponding state of $\mathcal{R}_A$ in a matrix $\mathbf{S}$ which is available to the followers. Since the robot has to be at a distance $d \cdot i$ to $\mathcal{R}_A$, it has to follow the position that $\mathcal{R}_A$ had at a time $t - \delta_i$ when it was at a distance $d \cdot i$ m earlier. Now, the speed to be followed is that of $\mathcal{R}_A$ at the current time t. Indeed, if $\mathcal{R}_A$ slows down, all followers should also slow down immediately. Thus the control to be given at time t is

$$\begin{aligned}\mathbf{u}(i) &= \begin{pmatrix} -v_i \sin\theta_i & \cos\theta_i \\ v_i\cos\theta_i & \sin\theta_i \end{pmatrix}^{-1} \\ &\quad \cdot \left(\begin{pmatrix} x_a(t-\delta_i) \\ y_a(t-\delta_i) \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix} + v_a(t) \cdot \begin{pmatrix} \cos(\theta_a(t-\delta_i)) \\ \sin(\theta_a(t-\delta_i)) \end{pmatrix} - \begin{pmatrix} v_i\cos\theta_i \\ v_i\sin\theta_i \end{pmatrix} \right)\end{aligned}$$

where $(x_i, y_i, \theta_i, v_i)$ is the state of the ith robot at time t. The values $(x_a(t-\delta_i), y_a(t-\delta_i), \theta_a(t-\delta_i))$ have been stored in the matrix $\mathbf{S}$. It was $\frac{d \cdot i}{ds}$ steps earlier. Figure 3 illustrates the behavior of the control law and shows that the distance between two robots does not depend on the position of the robots on the ellipse.

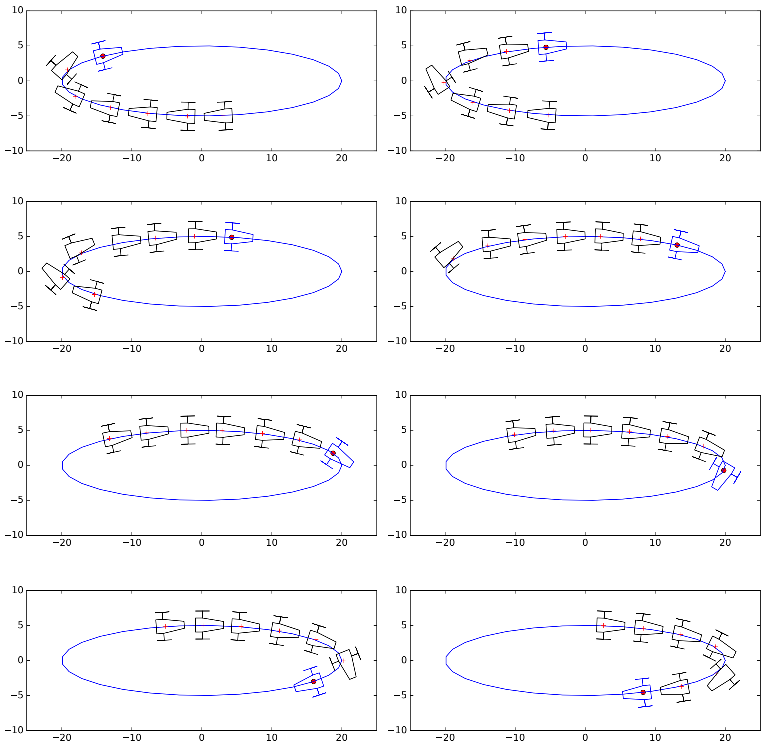

Figure 3: After the transient phase, all robots follow the leader with constant separation distance between two robots

3 Computing the secure zone

We consider n robots $\mathcal{R}_1, \dots, \mathcal{R}_n$ at positions $\mathbf{a}_1, \dots, \mathbf{a}_n$ in a 2D world $\mathbb{O}$. We assume that this group of robots move in order to build a barrier to protect a zone against the presence of any intruder, for instance, using the techniques presented in the previous section. We now want to provide a numerical set-membership method to guarantee that indeed there is no moving intruder which has entered inside a subzone of $\mathbb{O}$.

3.1 Complementary approach

We show here that the problem of computing the secure zone is equivalent (or more precisely complementary) to building a set-membership observer. For this purpose, we assume the following

- There exists a virtual intruder moving inside $\mathbb{O}$.
- The virtual intruder satisfying the differential inclusion[15]
$$\dot{\mathbf{x}}(t) \in \mathbb{F}(\mathbf{x}(t)),$$
where $\mathbf{x}(t)$ is its state vector.
- Each robot $\mathcal{R}_i$ has a visibility zone of the form $g_{\mathbf{a}_i}^{-1}([0, d])$ where d is the scope.

Our contribution is to show that characterizing the secure zone translates into a set-membership set estimation problem [16] where $\mathbf{x}(t)$ is shown to be inside the set $\mathbb{X}(t)$ returned by our set-membership observer. Then we conclude that $\mathbf{x}(t)$ cannot be inside the *complementary* of $\mathbb{X}(t)$. The complementary corresponds to the secure zone. This result can be formalized by the following theorem.

Theorem. The virtual intruder has a state vector $\mathbf{x}(t)$ inside the set

$$\mathbb{X}(t) = \mathbb{O} \cap (\mathbb{X}(t - dt) + dt \cdot \mathbb{F}(\mathbb{X}(t - dt)) \cap \bigcap_i g_{\mathbf{a}_i(t)}^{-1}([d(t), \infty]), \qquad (2)$$

where $\mathbb{X}(0) = \mathbb{O}$. As a consequence, the secure zone is

$$\mathbb{S}(t) = \overline{proj_{world}(\mathbb{X}(t))}. \qquad (3)$$

Proof (sketch). Two cases should be considered. If no actual intruder exists then $\mathbb{S}(t)$ cannot contain the intruder. If the virtual intruder is a real one, its state $\mathbf{x}(t)$ is inside $\mathbb{X}(t)$ and its position (which is a part of the state) is inside $proj_{world}(\mathbb{X}(t))$. In both situations, the intruder cannot be inside $\mathbb{S}(t)$.

3.2 Mathematical morphology

For the implementation of equation 2. We assume that the state of the intruder corresponds to its position. Thus the state space is of dimension 2 and corresponds to the 2D world. As a consequence, the projection operator in (3) is not needed anymore and $\mathbb{S}(t) = \overline{\mathbb{X}(t)}$. The set-membership evolution equation can be obtained using mathematical morphology [17] that is now recalled.

The Minkowski sum and the Minkowski difference of two sets $\mathbb{A}$ and $\mathbb{B}$ are defined by

$$\begin{array}{rcl} \mathbb{A}+\mathbb{B} & = & \{\mathbf{a}+\mathbf{b}\in\mathbb{R}^n | \mathbf{a}\in\mathbb{A} \text{ and } \mathbf{b}\in\mathbb{B}\} \\ \mathbb{A}-\mathbb{B} & = & \{\mathbf{a}-\mathbf{b}\in\mathbb{R}^n | \mathbf{a}\in\mathbb{A} \text{ and } \mathbf{b}\in\mathbb{B}\}. \end{array} \tag{4}$$

These operations correspond to the addition and difference used in interval computation. When $\mathbb{A}$ or $\mathbb{B}$ are singletons, we get these rules yield:

$$\begin{array}{rcll} \mathbb{A}+\mathbf{b} & = & \{\mathbf{a}+\mathbf{b}\in\mathbb{R}^n | \mathbf{a}\in\mathbb{A}\} & \text{(translation by } \mathbf{b}) \\ \mathbb{A}-\mathbf{b} & = & \{\mathbf{a}-\mathbf{b}\in\mathbb{R}^n | \mathbf{a}\in\mathbb{A}\} & \text{(translation by } -\mathbf{b}) \\ -\mathbb{A} & = & \{-\mathbf{a}\in\mathbb{R}^n | \mathbf{a}\in\mathbb{A}\}. & \text{(symmetry)} \end{array} \tag{5}$$

Dilatation. The *dilatation* of a set $\mathbb{A}$ by the *structuring element* $\mathbb{B}$ is defined by $\mathbb{A}+\mathbb{B}$.

Erosion. The *erosion* of $\mathbb{A}$ by $\mathbb{B}$ is defined by the three following equivalent relations:

$$\begin{array}{rcl} \mathbb{A}\ominus\mathbb{B} & = & \{\mathbf{z}\in\mathbb{R}^n | \mathbb{B}+\mathbf{z}\subseteq\mathbb{A}\} \\ & = & \bigcap_{\mathbf{b}\in\mathbb{B}} \mathbb{A}-\mathbf{b} \\ & = & \overline{\overline{\mathbb{A}}-\mathbb{B}}. \end{array} \tag{6}$$

where $\overline{\mathbb{X}}$ denotes the complement of $\mathbb{X}$ relative to $\mathbb{R}^n$.

Opening. The *opening* of $\mathbb{A}$ by $\mathbb{B}$ is obtained by the erosion of $\mathbb{A}$ by $\mathbb{B}$, followed by dilation by $\mathbb{B}$:

$$\mathbb{A}\circ\mathbb{B} = (\mathbb{A}\ominus\mathbb{B})+\mathbb{B} \tag{7}$$

Closing. The *closing* of $\mathbb{A}$ by $\mathbb{B}$ is obtained by the dilatation of $\mathbb{A}$ by $\mathbb{B}$, followed by erosion by $\mathbb{B}$:

$$\mathbb{A}\bullet\mathbb{B} = (\mathbb{A}+\mathbb{B})\ominus\mathbb{B} \tag{8}$$

Properties of the basic operators. Here are some properties of the basic binary morphological operators :

They are increasing, that is, if

$$\begin{array}{ccc} \mathbb{A} \subset \mathbb{C} & \Rightarrow & \mathbb{A} + \mathbb{B} \subset \mathbb{C} + \mathbb{B} \\ \mathbb{A} \subset \mathbb{C} & \Rightarrow & \mathbb{A} \ominus \mathbb{B} \subset \mathbb{C} \ominus \mathbb{B} \\ \mathbf{0} \in \mathbb{B} & \Rightarrow & \mathbb{A} \ominus \mathbb{B} \subseteq \mathbb{A} \circ \mathbb{B} \subseteq \mathbb{A} \subseteq \mathbb{A} \bullet \mathbb{B} \subseteq \mathbb{A} + \mathbb{B} \\ \mathbb{A} \subseteq (\mathbb{C} \ominus \mathbb{B}) & \Leftrightarrow & (\mathbb{A} \oplus \mathbb{B}) \subseteq \mathbb{C} \end{array} \tag{9}$$

Moreover

$$\begin{array}{ccc} \mathbb{A} + \mathbb{B} & = & \mathbb{B} + \mathbb{A} \\ (\mathbb{A} + \mathbb{B}) + \mathbb{C} & = & \mathbb{A} + (\mathbb{B} + \mathbb{C}) \\ (\mathbb{A} \ominus \mathbb{B}) \ominus \mathbb{C} & = & \mathbb{A} \ominus (\mathbb{B} + \mathbb{C}) \\ \mathbb{A} + \mathbb{B} & = & \overline{\overline{\mathbb{A}} \ominus -\mathbb{B}} \\ \mathbb{A} \bullet \mathbb{B} & = & \overline{\overline{\mathbb{A}} \circ -\mathbb{B}} \end{array} \tag{10}$$

4 Illustration

To illustrate our method, we consider that there exists an intruder in the set $\mathbb{O}$ which corresponds to the Bay of Biscay with an area of about 220 000 km^2. The set $\mathbb{O}$ is represented by Figure 4. All computations made in this section have been performed using contractor/separator algebra [18][19], except for the erosion and the dilatation that are taken from OpenCV.

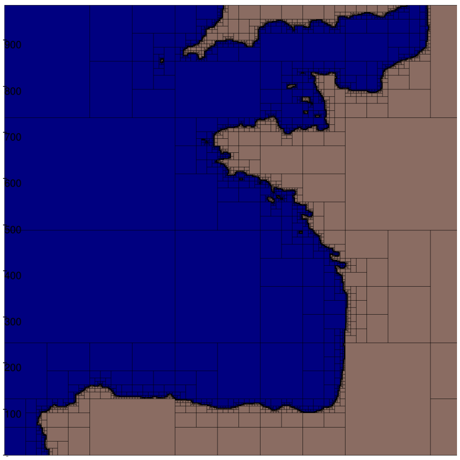

Figure 4: In blue is represented the set $\mathbb{O}$ which is supposed to enclose the virtual intruder

4.1 Computation of secure zone

Assume that each robot $\mathcal{R}_i$ is able to detect the intruder if the distance is less than a scope distance d. This means that a robot located at position $\mathbf{a}$

and equipped with a range sensor within a scope distance of d, detects that no intruder exists in the set $g_{\mathbf{a}}^{-1}([d,\infty])$, where

$$g_{\mathbf{a}}(\mathbf{x}) = \|\mathbf{x}-\mathbf{a}\| \notin [0,d]. \tag{11}$$

Figure 5 considers a situation with two robots. The blue zone of all sub-figures is proved to contain the intruder. It corresponds to the right term in (2). The first sub-figure corresponds to the static case, *i.e.*, it is given by the set:

$$\mathbb{O} \cap \bigcap_{i\in\{1,2\}} g_{\mathbf{a}_i}^{-1}([d,\infty]). \tag{12}$$

Assume now that the virtual intruder has a speed less than $\bar{v}$, *i.e.*, $\|\dot{\mathbf{x}}\| \leq \bar{v}$. The motion of the virtual intruder thus obeys to the following state equation:

$$\left\{ \begin{array}{lcl} \dot{x}_1 & = & v\cos\psi \\ \dot{x}_2 & = & v\sin\psi \end{array} \right. \tag{13}$$

where $v \leq \bar{v}$. This also means that, $\mathbb{F}(\mathbf{x})$ corresponds to a disk $\mathbb{D}(\mathbf{0},\bar{v})$ of center $\mathbf{0}$ and radius $\bar{v}$ and does not depend on $\mathbf{x}$. Again, since we assume that $\mathcal{R}_i$ detects nothing, we have $\|\mathbf{x}-\mathbf{a}_i\| > d$. For the initialization, we assume that $\mathbb{X}(0) \subset \mathbb{O} \subset [\mathbf{x}](0)$ where $[\mathbf{x}](0)$ is a box. Sub-Figures (b),(c),(d) of Figure 5 shows the corresponding secure zone painted magenta. The blue zone, contains the virtual intruder.

Figure 5: No intruder can be inside the magenta zone

4.2 Smoother

It is be possible to take into account the future to increase the secure zone. Indeed, it the intruder is at position $\mathbf{x}$, and if for all feasible maneuvers, the intruder will be detected in the future, then the point $\mathbf{x}$ belongs to what we call the *anti-causal secure zone*. The complete secure zone is thus composed of two regions that may overlap: the causal and the anticausal secure zones. Its complementary $\mathbb{X}$ can be computed by the following contractions

$$\begin{array}{rcl} \mathbb{X}(t) & = & \mathbb{X}(t) \cap \mathbb{O} \\ \mathbb{X}(t) & = & \mathbb{X}(t) \cap (\mathbb{X}(t-dt) + dt \cdot \mathbb{F}(\mathbb{X}(t-dt))) \\ \mathbb{X}(t) & = & \mathbb{X}(t) \cap \bigcap_i g^{-1}_{\mathbf{a}_i(t)}([d(t), \infty]) \\ \mathbb{X}(t) & = & \mathbb{X}(t) \cap \ (\mathbb{X}(t+dt) - dt \cdot \mathbb{F}(\mathbb{X}(t+dt))) \end{array}$$

with the initialization $\mathbb{X}(t) = \mathbb{R}^n$. The contractions should be performed up to the fixed point. This corresponds to a set-membership smoother. As shown in [20] the fixed point is reach in two steps: a forward step followed by a backward step.

4.3 Combining with a barrier strategy

Each robot follows a reference point. All reference points form a flat ellipsoid which plays the role of a barrier (see Section 2). The strategy is illustrated by Figure 1 for 10 robots. The set $\mathbb{O}$ corresponds to the blue area (left). On the right, $\mathbb{S}(t)$ is painted green. The observer has been implemented using interval analysis.

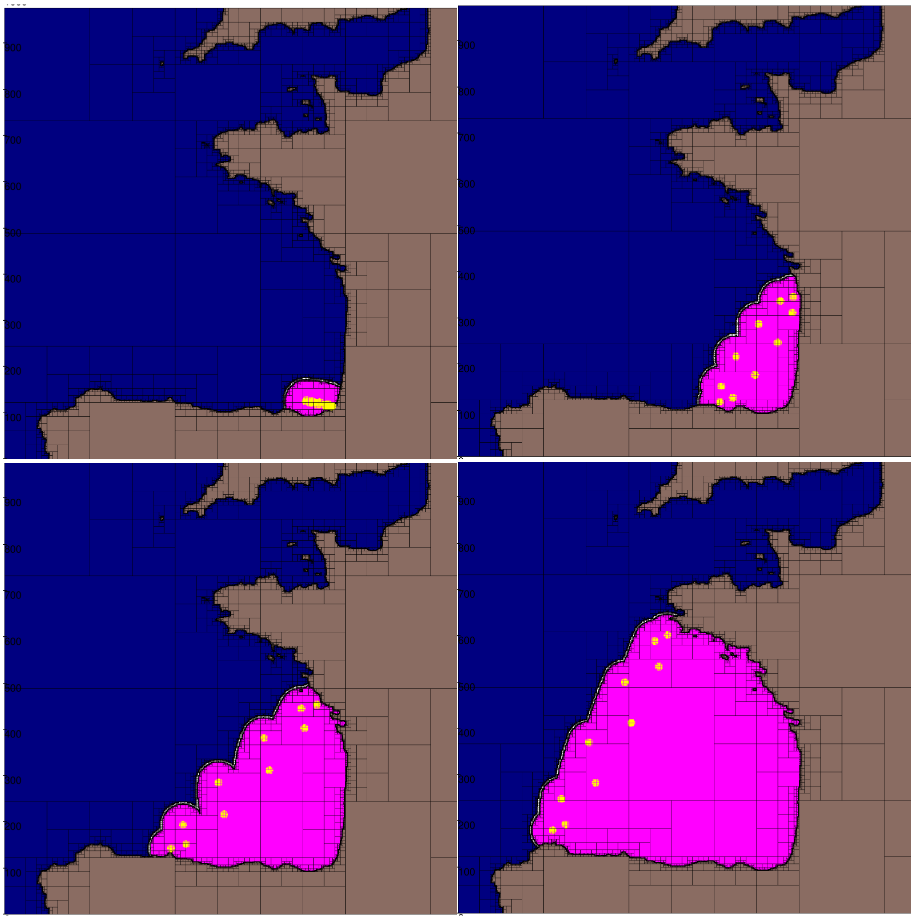

Figure 6: The ellipse moves in order to extend the secure zone (magenta). The position of the robots are represented by yellow points.

5 Conclusion

In this paper, we have proposed a set-membership method to compute the secure zone associated to a group of robots. In the secure zone, no undetected intruder can exist. We also proposed a control strategy to make the secure zone as large as possible with a limited number of robots. The principle is to make the group follow a flat ellipse which makes a barrier. Some perspectives to this work are the following

- Find a control strategy based on the shape of the secure zone and also on the shape of the boundary of $\mathbb{O}$. The ellipse strategy is indeed suited to convex shape for $\mathbb{O}$, but cannot deal with more complex shapes. For instance to secure the sea around an archipelago such as UK or Japon cannot be considered with a single barrier.

- Adapt the method to capture an intruder. If we assume that the intruder is captured as soon as it is detected, then we can guess that there exists strong links between securing a zone and the capture problem.

- Consider the escape problem. We are now on the side of the intruder. Escaping from capture by a group of robots or entering inside a zone without being detected can be considered a dual to the problem of securing a zone.

As previously said, we are not totally convinced by our control strategy which is only adapted to convex zones and is not efficient in case of zones with islands (such as archipelago). Since the basic model of Reynolds can been extended in several different ways to incorporate the effects of fear, emotions, leadership change, etc, we would like to take into account the shape of the current secure zone in our control strategy. For instance, we can add a law which motivates each robot to follow a leader (*chain building low*), and another low to follow the boundary of the computed secure zone (*boundary following low*). The strategy could also be take advantage of algorithms taken from operational research such as the solutions of the art gallery problems [21].

Acknowledgment. The authors want to thank the following students that where strongly involved in the project: Elouan Autret, Thomas Boulier, Maxime Bouyssou, Philippe Chuzel, Alice Danckaers, David Duverger, Raphael Finkelstein, Sylvain Hunault, Pierre Jacquot, Mael Le Gallic, Eric Mourre, Thiago Oliveira, Benoit Raymond, Khadimoullah Vencatasamy. They made a small video illustrating the project: https://youtu.be/rNcDW6npLfE.

References

[1] L. Jaulin, Mobile Robotics, ISTE editions, 2015.

[2] V. Creuze, Robots marins et sous-marins ; perception, modélisation, commande, Techniques de l'ingénieur.

[3] M. Carreras, P. Ridao, R. García, T. Nicosevici, Vision-based localization of an underwater robot in a structured environment, in: Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, Vol. 1, IEEE, 2003, pp. 971–976.

[4] S. Bazeille, V. Lebastard, F. Boyer, Different use of electric sense for underwater robots navigation, in: Ocean Engineering and Oceanography, Springer, Brest, France, 2017.

[5] F. Boyer, P. B. Gossiaux, B. Jawad, V. Lebastard, M. Porez, Model for a sensor inspired by electric fish, IEEE transactions on robotics 28 (2) (2012) 492–505.

[6] B. Zerr, G. Mailfert, A. Bertholom, H. Ayreault, Sidescan sonar image processing for auv navigation, in: Oceans 2005-Europe, Vol. 1, IEEE, 2005, pp. 124–130.

[7] A. Caiti, A. Garulli, F. Livide, D. Prattichizzo, Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach, IEEE Journal of Oceanic Engineering 30 (1) (2005) 140–152.

[8] A. Caiti, A. Garulli, F. Livide, D. Prattichizzo, Set-membership acoustic tracking of autonomous underwater vehicles, Acta Acustica united with Acustica 5 (88) (2002) 648–652.

[9] F. L. Bars, J. Sliwka, O. Reynet, L. Jaulin, State estimation with fleeting data, Automatica 48 (2) (2012) 381–387.

[10] L. Jaulin, Automation for Robotics, ISTE editions, 2015.

[11] H. G. Tanner, A. Jadbabaie, G. J. Pappas, Stable flocking of mobile agents, part i: Fixed topology, in: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, Vol. 2, IEEE, 2003, pp. 2010–2015.

[12] E. Rimon, D. E. Koditschek, Exact robot navigation using artificial potential fields, "IEEE Transactions on Robotics and Automation" 8 (5) (1992) 501–518.

[13] C. Petres, M. R. Ramirez, F. Plumet, Reactive Path Planning for Autonomous Sailboat, in: IEEE International Conference on Advanced Robotics, 2011, pp. 1–6.

[14] C. Pêtrès, M.-A. Romero-Ramirez, F. Plumet, A potential field approach for reactive navigation of autonomous sailboats, Robotics and Autonomous Systems 60 (12) (2012) 1520–1527.

[15] J.-P. Aubin, Viability Theory, Springer Science and Business Media, 2009.

[16] M. Kieffer, L. Jaulin, E. Walter, D. Meizel, Nonlinear identification based on unreliable priors and data, with application to robot localization, in: A. Garulli, A. Tesi, A. Vicino (Eds.), Robustness in Identification and Control, Vol. LNCIS 245, Springer-Verlag, London, UK, 1999, pp. 190–203.

[17] L. Najman, H. T. (Eds), Mathematical morphology: from theory to applications, ISTE-Wiley, 1000.

[18] G. Chabert, L. Jaulin, Contractor Programming, Artificial Intelligence 173 (2009) 1079–1100.

[19] L. Jaulin, B. Desrochers, Introduction to the algebra of separators with application to path planning, Engineering Applications of Artificial Intelligence 33 (2014) 141–147.

[20] L. Jaulin, M. Kieffer, I. Braems, E. Walter, Guaranteed nonlinear estimation using constraint propagation on sets, International Journal of Control 74 (18) (2001) 1772–1782.

[21] J. O'Rourke, Art Gallery Theorems and Algorithms, Oxford University Press, 1987.