Go Zone

1 Problem

We have an underwater robot with a torpedo shape, for instance a Riptide or a Folaga from Kopadia. It is equipped with an IMU and a pressure sensor. The only exteroceptive sensor is an echo-sounder, on the nose of the AUV, as illustrated by Figure 1. It has no other sensors that can be used for localisation such as cameras or lateral sonars. Note that the robot also contains a GPS and a Wifi connection. Now, since the robot is always underwater, they will not be used.

At ENSTA, we have 4 riptides and a collaboration between AUVs can be thought. Now, since they do not contain any communication system that could work underwater, no collaboration will be considered, at least in the initial part of the project.

The problem we want to consider is now described. We have have an underwater environment with a know bathymetry. We want to find several zones in the world, so that the AUV is able to move from zones to zones without getting lost. The AUV is assumed to move horizontally at a constant depth.

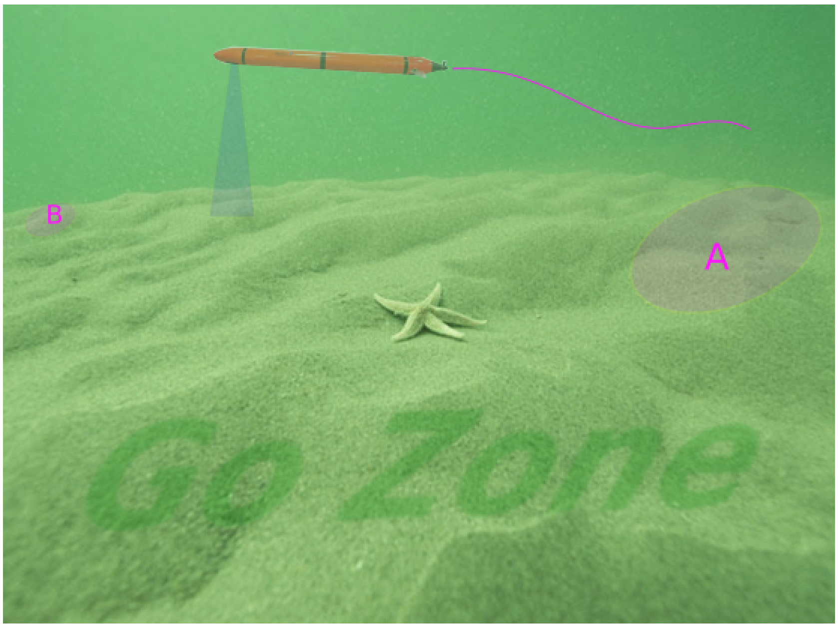

Figure 1: Illustration of the *Go Zone* project

Note that the robot has no bathymetric map inside its memory. It only contains a small automaton that described its behaviour.

2 Stable cycle

We want to explore an unknown environment, without any localization system and without being lost. By being lost, we mean not being able to reach a target set, or equivalently not being able to come back home. Under the water, we can easily know the depth using a barometer and the problem of finding a path can be considered in the horizontal 2D plane. If we are able to measure some quantities such as the altitude, the temperature, the salinity, etc, we can find a reliable path which allows us not being lost. This is the case of underwater animals such as marine turtles, or whales which follow isotherms [2] with the help of an internal compass. These underwater animals do not know where they are, do not have in memory a map of the environment. However, from the evolutionary process, they know how to perform a stable cycle.

Figure 2 illustrates what is a stable cycle. We have a ball entering inside the blue room with the heading $\bar{\phi}_1$. When the ball bounces the ith wall, it take the heading $\bar{\psi}_i$. We can understand that we converge to a stable cycle.

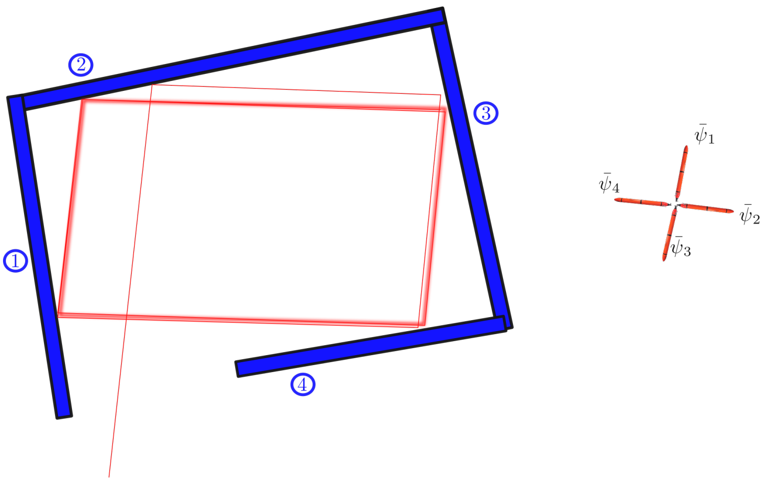

Figure 2: The bouncing ball converges to a stable cycle

The same principle can be performed with an AUV, as illustrated by Figure 2 moving at a constant depth. The AUV is able to create an event when it crosses upward an isobath at 10m. Taking the corresponding headings $\bar{\psi}_i$ we may obtain a stable cycle. Note that there is no need for the AUV to know the bathymetry of the world.

The corresponding automaton to which the robot has to obey is illustrated by Figure 2. For instance, assume that the AUV starts in the polygon $\mathbb{A}$ and follows the heading $\bar{\psi}_1$ then, it is in the state denoted by $\mathbb{A} \to \mathbb{B}$. When it detect the isobath of -10m, then it knows that it is inside $\mathbb{B}$. More precisely, it knows that it is at the intersection of the -10m isobath and the zone $\mathbb{B}$. In this case, it switches to the automaton state $\mathbb{B} \to \mathbb{C}$ and follows the heading $\bar{\psi}_2$. If the AUV continues to obey to the automaton, it will perform a stable cycle.

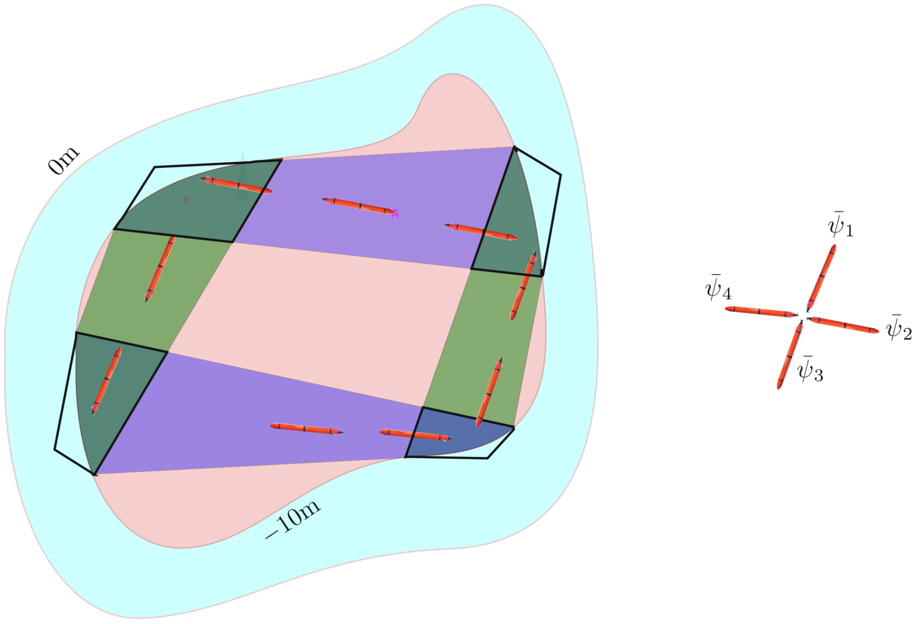

Figure 3: Stable circuit made by the AUV

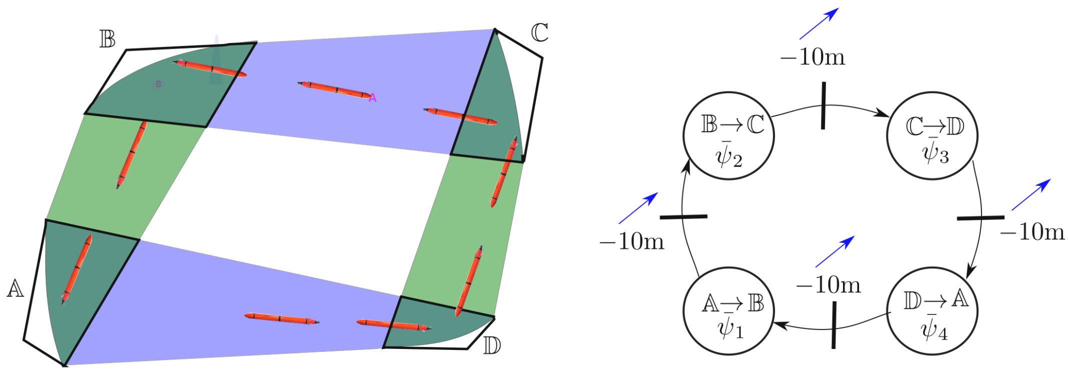

Figure 4: Pétri net associated to the circuit

3 Supervision

Assume that we are able to get several circuits as illustrated by the Pétri graph of Figure 3. At the red transitions, we are able to choose between different states. Therefore, we can choose which zone we want to visit. We can also choose the circuit we want to follow. The circuits are $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{A}$ or $\mathbb{E}, \mathbb{C}, \mathbb{F}, \mathbb{D}, \mathbb{E}$ but can be more complex, such as $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{E}, \mathbb{C}, \mathbb{D}, \mathbb{A}$.

4 Model of the AUV

For the simulation, but also for the state estimation, we need a model for the AUV. We recall the mechanisation kinematic equation of a solid body

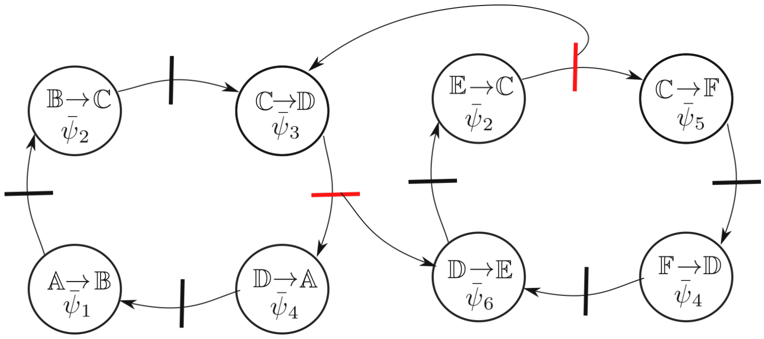

Figure 5: Pétri net with several circuits

$$\left\{\begin{array}{ll}\dot{\mathbf{p}} & = \mathbf{R}(\varphi,\theta,\psi)\ \cdot\ \mathbf{v}_r \\ \dot{\mathbf{v}}_r & = \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \begin{pmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix} & = \begin{pmatrix}1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \frac{\sin\varphi}{\cos\theta} & \frac{\cos\varphi}{\cos\theta}\end{pmatrix}\cdot\boldsymbol{\omega}_r\end{array}\right. \tag{1}$$

Torpedo assumptions. We assume that

- *(i)* we have no *side slip* effect (i.e., $\mathbf{v}_r$ has the direction of the AUV);
- *(ii)* from the friction and propulsion point of view, the AUV goes straight.

No side slip simplification. The assumption of no side slip means that $\mathbf{v}_r = (v_{rx}, v_{ry}, v_{ry}) = (v, 0, 0)$. Therefore, we get (1) becomes

$$\left\{\begin{array}{ll}\dot{\mathbf{p}} & = \mathbf{R}(\varphi,\theta,\psi)\ \cdot\ (v\,0\,0)^{\mathrm{T}} \\ \dot{v} & = a_{rx} \\ \begin{pmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix} & = \begin{pmatrix}1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \frac{\sin\varphi}{\cos\theta} & \frac{\cos\varphi}{\cos\theta}\end{pmatrix}\cdot\boldsymbol{\omega}_r\end{array}\right.$$

This model is kinematic and no dynamical coefficients such as the mass, the inertia, frictions appear. As illustrated by Figure 4 using the three fins and the propellers, we can control $\boldsymbol{\omega}_r$ and a_{rx} respectively.

Newton low with the added mass. Taking into account the assumption (ii), we get that

$$f_x = (m + m_a)a_{rx}$$

where m_a is the added mass. We have

$$m_a = k_a \rho_{water} V$$

where V is the volume of the AUV and k_a is a coefficient which depends of the shape of the robot. Since our AUV has a cylinder shape, we can take $k_a = 1$. Moreover, since the buoyancy of the AUV is almost neutral, we have $\rho_{water} V \simeq m$. Thus, $m \simeq m_a$. Moreover, we have

$$f_x = k_p \cdot u_0 \cdot |u_0| - \frac{1}{2} C_x S_x \rho_{water} v \cdot |v|$$

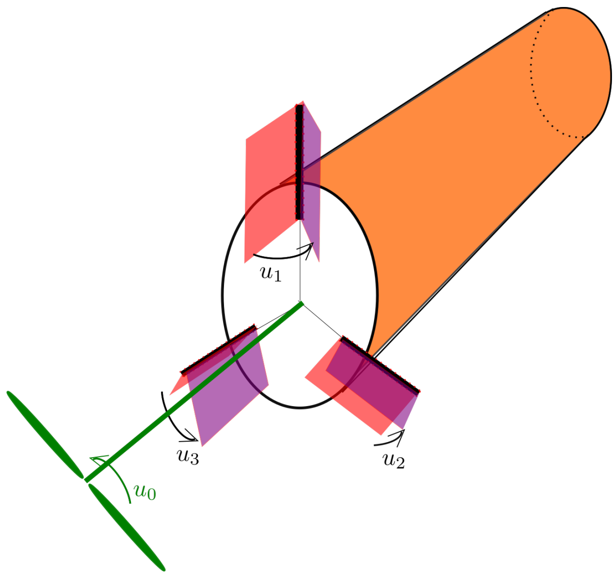

Figure 6: Building the model of the Riptide

where u_0 is the rotation speed to the propeller, or equivalently the voltage of the motor feeding the propeller. Finally, we get

$$a_{rx} = \frac{f_x}{m + m_a} = p_1 u_0 \cdot |u_0| - p_2 v \cdot |v|.$$

All coefficients $V, k_a, \rho_{water}, k_p, C_x, S_x$ are approximately known. Now, they can be expressed with two parameters only p_1, p_2 which can be identified by some experiments.

Rotations

Each fin generate a drag and a lift. The drag can be neglected, but not the lift which is used to control the rotation of the AUV. We take the following model :

$$\boldsymbol{\omega}_r = \begin{pmatrix} \omega_{rx} \\ \omega_{ry} \\ \omega_{rz} \end{pmatrix} = v \cdot \underbrace{\begin{pmatrix} p_3 & 0 & 0 \\ 0 & p_4 & 0 \\ 0 & 0 & p_4 \end{pmatrix} \cdot \begin{pmatrix} -1 & -1 & -1 \\ 0 & \sin\frac{2\pi}{3} & -\sin\frac{2\pi}{3} \\ -1 & \cos\frac{2\pi}{3} & -\cos\frac{2\pi}{3} \end{pmatrix}}_{\mathbf{B}} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

This formula tells us that each fin contributes in the same way to the roll ω_{rx} and that the fin 1 does not contribute to the pitch.

State equation

The state equation of the Riptide are the following

$$\left\{ \begin{array}{ll} \dot{\mathbf{p}} & = \mathbf{R}(\varphi, \theta, \psi) \;\cdot\; (v\,0\,0)^{\mathrm{T}} \\ \dot{v} & = p_1 u_0 \cdot |u_0| - p_2 v \cdot |v| \\ \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} & = \begin{pmatrix} 1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \frac{\sin\varphi}{\cos\theta} & \frac{\cos\varphi}{\cos\theta} \end{pmatrix} \cdot v \cdot \mathbf{B}(p_3, p_4) \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \end{array} \right. \tag{2}$$

where

$$\mathbf{B}(p_3,p_4)=\begin{pmatrix} p_3 & 0 & 0\\ 0 & p_4 & 0\\ 0 & 0 & p_4\end{pmatrix}\cdot\begin{pmatrix} -1 & -1 & -1\\ 0 & \sin\frac{2\pi}{3} & -\sin\frac{2\pi}{3}\\ -1 & \cos\frac{2\pi}{3} & -\cos\frac{2\pi}{3}\end{pmatrix}$$

The inputs are u_0, u_1, u_2, u_3.

5 Identification

The parameter vector $\mathbf{p}=(p_1,p_2,p_3,p_4)^{\mathrm{T}}$ has to be estimated using some basic experiments.

5.1 Estimation of p_2

To identify p_2 we propose an experiment where the AUV goes straight with no propulsion in a pool. With a camera, we localise the robot at different instants. We have

$$\dot{v}=-p_2 v^2.$$

The solution is

$$v(t)=\frac{v(0)}{v(0)p_2 t+1}.$$

The corresponding position is

$$x(t)=x(0)+\int_0^t \frac{v(0)}{v(0)p_2 t+1}dt=x(0)+\frac{\log\left(v(0)p_2 t+1\right)}{p_2}$$

Using SIVIA, from measurement of different position, we can identify both $v(0)$ and p_2.

5.2 Estimation of p_1

We assume that p_2 has been identified. With the AUV, we go straight with the propeller on and u_0 constant. Once $\dot{v}=0$, we have

$$p_1 u_0^2 - p_2 v^2 = 0.$$

Thus

$$p_1 = p_2\frac{v^2}{u_0^2}.$$

6 Control

For small angles of the fins, $\sin u_i \sim u_i$. If we want a rotation vector $\bar{\boldsymbol{\omega}}_r$, we may take

$$\mathbf{u}=\frac{1}{v}\mathbf{B}^{-1}\bar{\boldsymbol{\omega}}_r.$$

For the control, we may need an estimation of the position and of the speed of the AUV. To get this estimation, we integrate the two first equations of (2).

7 Tubes

7.1 Circular path

For some situations, a circular path is needed as explained in Figure 7.1. Again, we can get a stable cycle. The circular path will be computed using tubes.

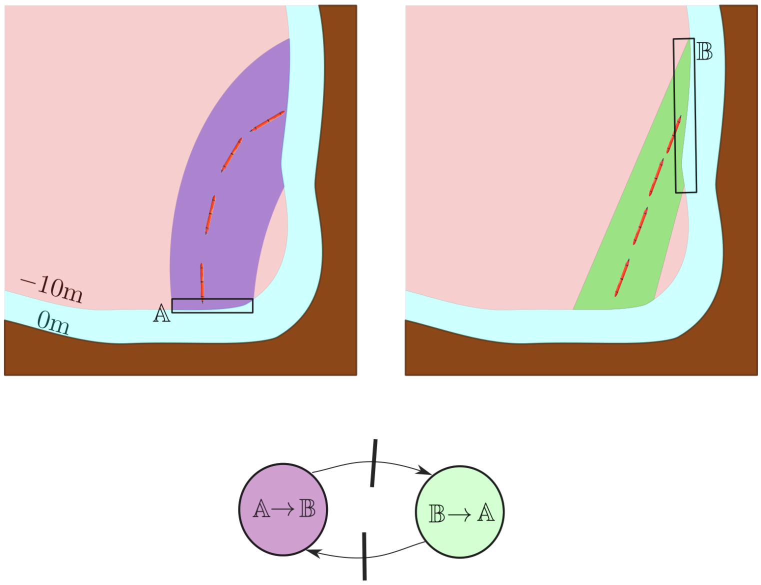

Figure 7: Circular path combined with a linear path

7.2 Representation of tubes

A tube is a function which associates to any $t \in \mathbb{R}$ a subset of $\mathbb{R}^n$. In the case where these subsets are intervals or boxes, a tube can be represented in the computer by stepwise functions (see [1]) as illustrated in Figure 8.

References

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

[2] K. Lohmann and C. Lohmann. Orientation and open-sea navigation in sea turtles. *Journal of Experimental Biology*, 00(199):73–81, 1996.

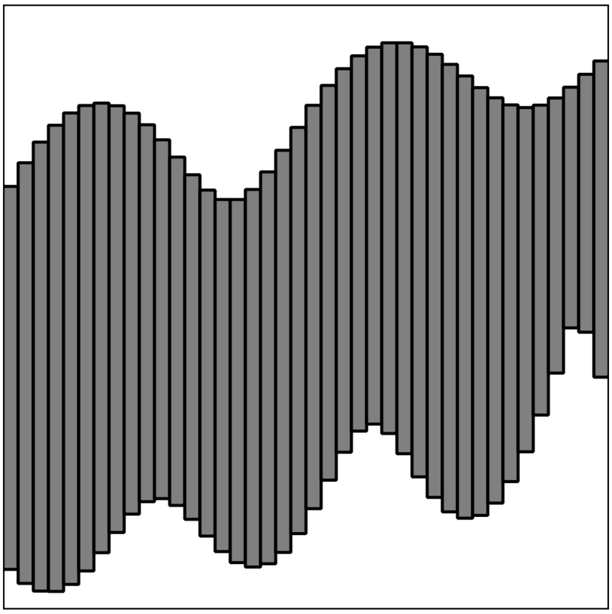

Figure 8: In numerical computations, a tube $[f](t)$ can be approximated by a lower and an upper stepwise functions $f^{-}(t)$ and $f^{+}(t)$. The tube $[f](t)$ encloses an uncertain trajectory $f(t)$