

# COMPTE RENDU RIPTIDE – 16/11/2020

## Contrôleur :

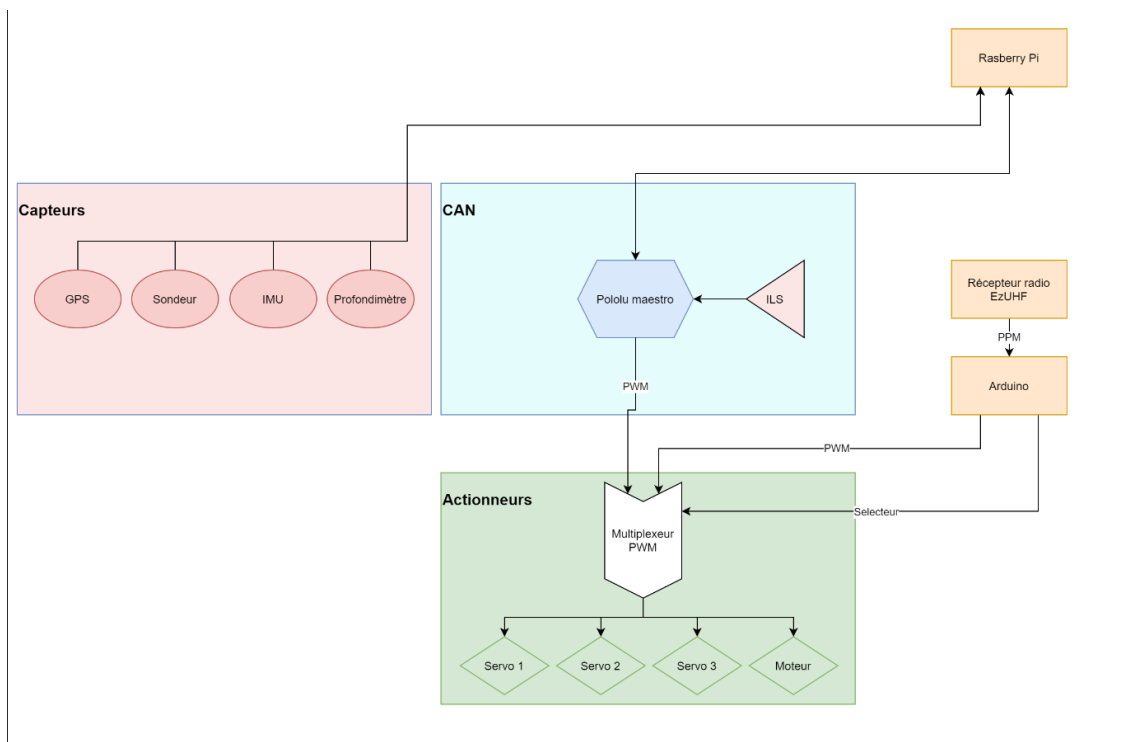
- **Fait :**
  - Prise en main du code existant et passage sous python3 ;
  - Contrôleur python avec suivi d'une liste de waypoints en partie sous ROS.
- **Reste à faire :**
  - Réel planificateur de mission (utiliser QGroundControl par exemple) ;
  - Finaliser le passage sous ROS afin d'utiliser correctement le contrôleur sur le robot et sur gazebo.

## Simulation Gazebo – UUV Simulator :

- **Fait :**
  - Simulation d'un AUV quelconque sur Gazebo ;
  - Ajout de capteurs sur Gazebo ;
  - Ajout de thrusters sur Gazebo ;
  - Modélisation 3D du Riptide (corps et ailerons) sur CAO.
- **En cours :**
  - Ajout d'ailerons ;
  - Déterminer les masses (matrices d'inerties, traînées).



## Electronique :



- **Architecture matérielle** : pour effectuer des tests sur la partie matérielle, nous allons ajouter une télécommande pour piloter le robot. Comme les récepteurs ne nous permettent pas d'avoir suffisamment de voies pour contrôler l'ensemble des actionneurs (+sélecteur), nous allons rajouter une arduino (pour utiliser les 12 voies du PPM). Ceci a l'avantage de permettre un contrôle plus facile et intuitif pour commander le Riptide en nous permettant de mixer plus facilement les voies.  
Pour démarrer/arrêter le système, nous partons sur un interrupteur magnétique ILS.
- **Commande du matériel** : mail envoyé à M.Le Bars et M.Fourniol.
- **La suite** :
  - Organiser des séances à l'ENSTA (autorisation ?) pour avancer sur la préparation du robot.

# Rapport MagMap

## Création du pattern pour l'exploration de zone

Elèves : Kévin AFFRAIX et Colin BAUMGARD

Séance : 16/11/12

### **Colin :**

On souhaite déterminer la trajectoire que doit suivre le robot pour que la luge réalise un boustrophédon.

Deux méthodes ont été envisagées, et devront être testées pour déterminer la plus adaptée.

La première méthode consiste à remplacer chaque waypoint de la luge par deux wp robots : un à une distance  $l$  du wp courant dans l'axe du wp précédent et le wp courant, et l'autre à une distance  $l$  du wp courant et dans l'axe wp courant/suivant.

La deuxième méthode consiste à prendre la tangente en tout point de la courbe de la trajectoire luge et de mettre le robot à une distance  $l$  sur celle-ci.

### **Kévin :**

Travail terminé : On souhaite générer les points délimitant le boustrophédon de direction optimale sur une surface quelconque, modélisée par un polygone 2D. Le calcul de la direction optimale «  $\theta$  » de parcours d'un boustrophédon est réalisé à partir d'un algorithme exploitant la propriété convexe d'un polygone pour obtenir un résultat en complexité linéaire sur le nombre de points du polygone (*rotating calipers*). Une fois cela fait, il suffit d'intersecter le polygone avec des droites parallèles de direction  $\theta$  espacées d'une largeur de fauchée «  $d$  » à définir.

Prochain Jalon : La surface de travail n'étant pas forcément convexe, il faut maintenant implémenter un algorithme permettant de décomposer un polygone quelconque (avec éventuellement des polygones internes modélisant des obstacles) en plusieurs sous-parties convexes. Pour obtenir une forme s'adaptant bien aux boustrophédons, une décomposition trapézoïdale me semble la meilleure. Pour cela, il existe des algorithmes de type « sweep line » qu'il faut maintenant implémenter.

# Magnétomètres

Robin Agathe, Alexandre

Pour le magnétomètre :

## Partie hardware :

- Prendre en mains le magnétomètre : FAIT
- Interfacer le magnétomètre avec Linux : ECHEC, le logiciel semble ne fonctionner que sous Windows
- Tester le magnétomètre avec la luge : trop d'interférences et de bruit, calibration nécessaire
- Calibrer un magnétomètre d'IMU : FAIT
- Détecter un objet avec un magnétomètre d'IMU : FAIT
- Pousser la détection avec le magnétomètre d'IMU : à faire d'ici fin novembre
- Réaliser une analyse type Fourier avec le magnétomètre d'IMU : à faire d'ici fin novembre
- Calibrer le magnétomètre : à faire après le déconfinement (nécessite de sortir de la ville pour éviter les interférences)
- Détecter un objet avec le magnétomètre : après la calibration

## Partie simulation :

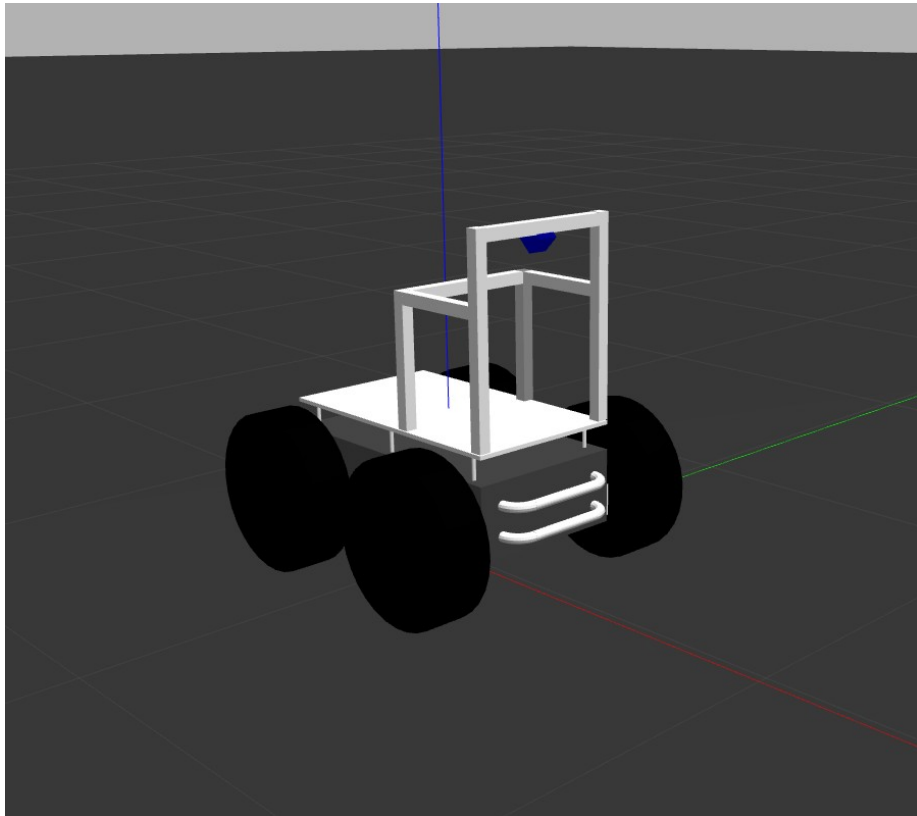
- Trouver une bibliothèque Python pour simuler le magnétomètre et l'environnement magnétique : FAIT
- Prendre en main la bibliothèque : en cours (mi-décembre)
- Comprendre les principes physiques du magnétomètre : FAIT
- Approfondir les principes physiques : en cours (début décembre)
- Tester le modèle python sur différents scénarios (forme de l'objet, profondeur, ...) et déterminer quelles caractéristiques offrent une bonne détection : (fin décembre)

## Partie cartographie :

- Analyser des données magnétiques et identifier les points d'intérêts : fin décembre
- Réaliser une carte magnétique : ?

## Simulation du robot Saturne

Avancement dans la simulation Gazebo, le modèle du robot s'affiche correctement dans gazebo et les roues tournent avec des commandes ROS. Cependant les roues tournent mais le robot n'avance pas. J'ai essayé de modifier les coefficients de frottements des roues sur le sol mais le problème ne vient pas de là. Il semble que le robot est fixe, même en appliquant des forces je ne peux pas le bouger.



## **Groupe : Prise en main de Saturne**

Dans le cadre de la prise en main du robot Saturne nous avons fait la mise en service du robot et nous avons testé les codes existants.

De plus nous avons commencé à adapter les codes de Saturne afin de pouvoir charger une mission quelconque à partir d'un fichier. Le fichier devra comporter ligne par ligne les étapes de la mission. Le format exacte du fichier de mission reste à déterminer et pourra éventuellement être adapté au besoin des autres groupes.

Durant les prochaines séances nous devons tester les codes adaptés sur le vrai robot et enregistrer les données de localisation du robot et de la luge pour le groupe qui travail sur la localisation de la luge.

# Magnetic mapping using a trailed magnetometer

Quentin Brateau<sup>1</sup>✉

<sup>1</sup>ENSTA Bretagne, Brest

The realization of a magnetic map of a terrain is a powerful tool especially used in mine warfare. This cartography can be done with a magnetometer, but the task is quite difficult. This is why it is useful to use robots to make this cartography. The problem induced by this solution is the addition of magnetic disturbances related to the structure of the robot and its actuators. It is then possible to drag the magnetometer on a sled behind the robot.

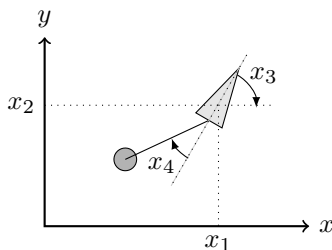
Magnetic | Mapping | Control | Interval Analysis  
Correspondence: [quentin.brateau@ensta-gretragne.org](mailto:quentin.brateau@ensta-gretragne.org)

## Modelisation of the system

The system is composed of a vehicle that will drag a sledge that transports a magnetometer. The towing vehicle is a tank type vehicle, and the sled is attached to it with a rope. Assuming that the rope is constantly under tension, we are able to find the equations describing the dynamics of the system. By noting  $x$  the state of the system and  $u$  its inputs, the dynamics of the system is described by :

$$\dot{x} = f(x, u) = \begin{cases} \dot{x}_1 = \frac{u_1 + u_2}{2} \cdot \cos(x_3) & (1) \\ \dot{x}_2 = \frac{u_1 + u_2}{2} \cdot \sin(x_3) & (2) \\ \dot{x}_3 = u_2 - u_1 & (3) \\ \dot{x}_4 = -\frac{u_1 + u_2}{2} \cdot \sin(x_4) - \dot{x}_3 & (4) \end{cases}$$

Here  $x_1$ ,  $x_2$  and  $x_3$  are respectively the abscissa, the ordinate and the heading of the trailing robot,  $x_4$  is the angle of the sled to the tractor vehicle.



## Simulator

A python simulation was realized using VIBES in order to test the behavior of the system. A class *Tank* has been created to instantiate a vehicle with its sled. Then the script integrates the evolution equation using Euler's method in order to obtain the state of the system  $x$  according to the  $u$  inputs.

We could see that the behavior of the system seems correct and the model is faithful to reality. Moreover, the GNSS sensor and an accelerometer are simulated in order to enclose the real position in a box.

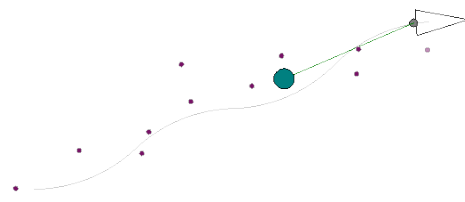


Fig. 1. Simulation of the system

## Reliable set of sled's angle

Considering that the rope remains continuously under tension while the robot is moving, then the evolution of the angle  $x_4$  of the sled relative to the towing vehicle is described by the *Differential Equation 4*.

We are able to find the trajectory of the sled by computing the interval containing the angle  $x_4$ , considering that initially  $x_4$  belongs to  $[-\pi/2; \pi/2]$ . Then by applying the *Differential Equation 4* on this interval, knowing the control vector  $u$ , we end up obtaining a fine interval framing the real angle  $x_4$ , independently of the initial angle as we can see on the FIGURE 2.

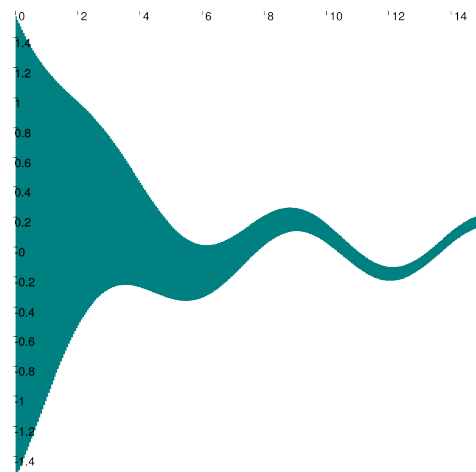


Fig. 2. Integration example using intervals

## Sled localization

By having a box enclosing the trailing robot, and an interval framing the angle  $x_4$ , we are able to obtain a box containing the sled, knowing the length of the rope.

This box has the shape of a pie sector. *Equation 5* presents the position of the magnetometer as a function of the state of the system  $x_1$ ,  $x_2$  and  $x_4$ . Here all these variables are intervals. By applying a polar contractor to these intervals, we are able to obtain the intervals containing  $x_m$  and  $y_m$ .

$$\begin{cases} x_m = x_1 - L.\cos(x_4) \\ y_m = x_2 - L.\sin(x_4) \end{cases} \quad (5)$$