Projet Go Zone

UV 5.7 - ARCHITECTURE ROBOTIQUE

SOUTENANCE

ENSTA Bretagne

Contexte

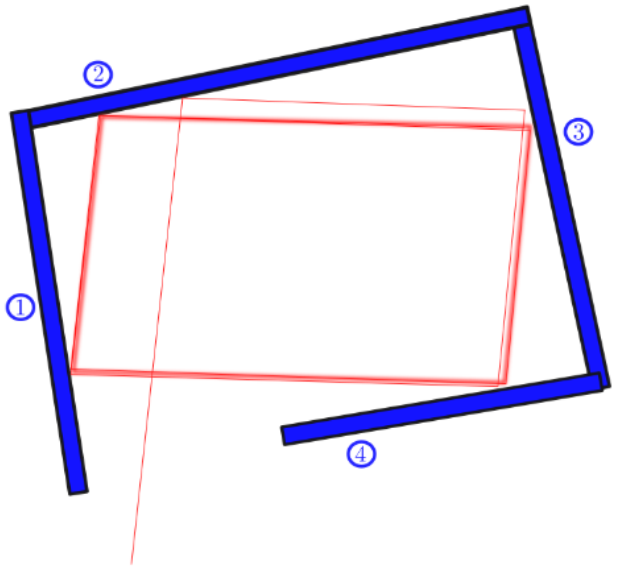
- Travail de groupe dans le cadre de l'UV 5.7 Architecture Robotique
- 11 séances - 56 heures
- Projet:
  - Nouvelle méthode de navigation sur zone pour AUV

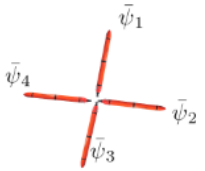
    Cycle stable :

    Consignes = Liste de caps

    Transitions = Valeur d'un isobathe

  - Preuve de concept pour la recherche de la Cordelière
  - AUV à disposition: Riptide et Folaga





Sommaire

I. Identification des tâches à réaliser

1. Tâches à réaliser
2. Interdépendances des tâches
3. Attribution des tâches
4. Agencement dans cette présentation

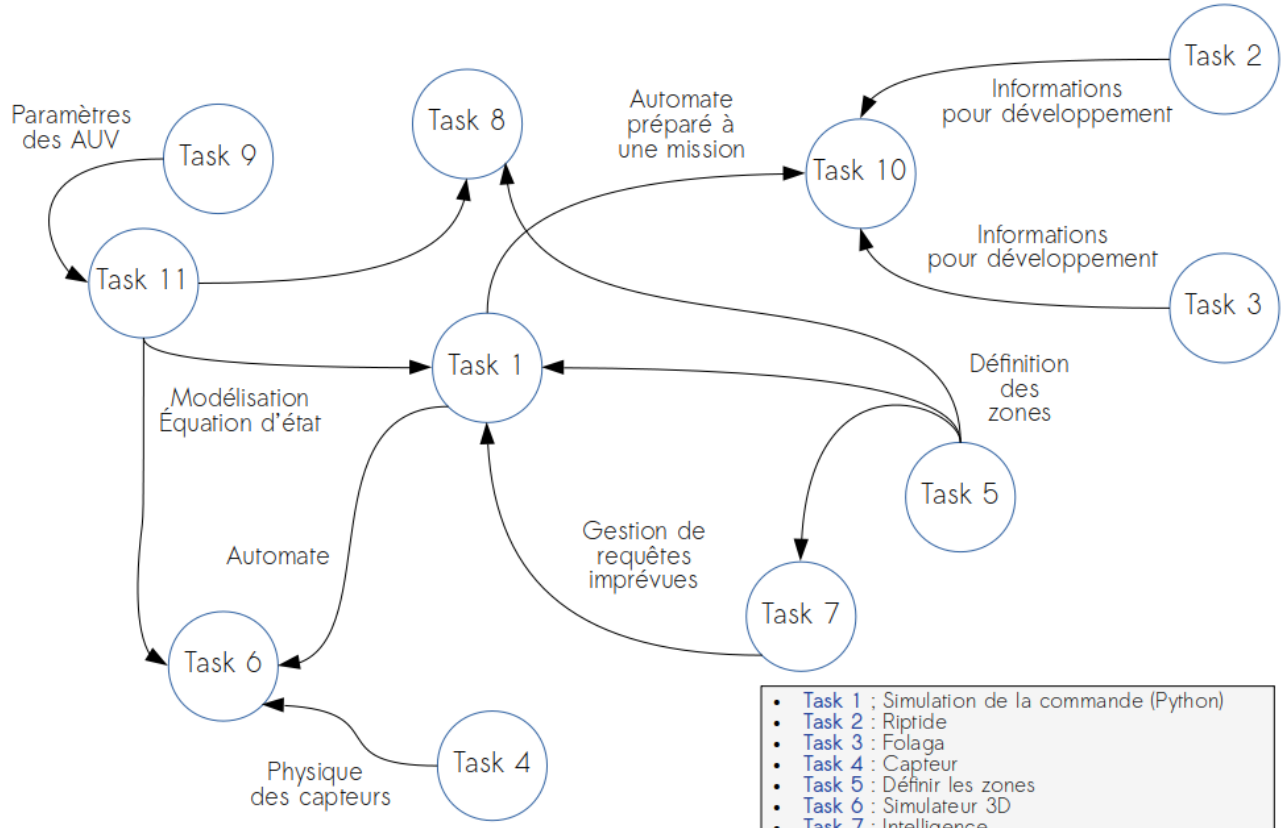
I. Identification des tâches à réaliser

1. Tâches à réaliser

Tâche	Description
1	Simulation de la commande
2	Utilisation des Riptide
3	Utilisation du Folaga
4	Gestion des capteurs
5	Définition des cycles stables
6	Simulateur 3D
7	Intelligence de navigation entre cycles stables
8	Tubex
9	Identification des paramètres des AUV
10	ROS / MOOS
11	Modélisation des AUV
12	Préparation de l'expérimentation
Architecte	Gestion du groupe

I. Identification des tâches à réaliser

2. Interdépendances des tâches



I. Identification des tâches à réaliser

3. Attributions des tâches

Tâche	Description	Membres
1	Simulation de la commande	Matthieu Bouveron, Clément Bichat
2	Utilisation des Riptide	Philibert Adam, Nathan Fourniol, Alexandre Courjaud
3	Utilisation du Folaga	Corentin Jegat, Yann Musellec
4	Gestion des capteurs	Philibert Adam, Matthieu Bouveron, Driss Tayebi
5	Définition des cycles stables	David Brellmann, Aurélien Lebrun, Alexandre Argento
6	Simulateur 3D	Cyril Cotsaftis, Quentin Cardinal
7	Intelligence de navigation entre cycles stables	Anouar Mahla, Axel Porlan
8	Tubex	Aurélien Grenier, Driss Tayebi
9	Identification des paramètres des AUV	Erwann Landais, Yann Musellec
10	ROS / MOOS	Nathan Fourniol, Corentin Jegat, Alexandre Courjaud, Philibert Adam
11	Modélisation des AUV	Quentin Cardinal, Cyril Cotsaftis, Erwann Landais
12	Préparation de l'expérimentation	Kévin Bedin
Architecte	Gestion du groupe	Kévin Bedin

I. Identification des tâches à réaliser

4. Agencement dans cette présentation

Tâche	Description
1	Simulation de la commande
2	Utilisation des Riptide
3	Utilisation du Folaga
4	Gestion des capteurs
5	Définition des cycles stables
6	Simulateur 3D
7	Intelligence de navigation entre cycles stables
8	Tubex
9	Identification des paramètres des AUV
10	ROS / MOOS
11	Modélisation de l'AUV
12	Préparation de l'expérimentation

Tâche	Partie
2, 4 & 10 (Riptide)	**II.1.a** Utilisation du Riptide
3, 4 & 10 (Folaga)	**II.1.b** Utilisation du Folaga
11	**II.2.a** Modèles des AUV
9	**II.2.b** Identification des paramètres des AUV
5	**II.3.a** Définition des cycles stables
7	**II.3.b** Intelligence de navigation entre cycles stables
8	**II.3.c** Tubex
1	**II.4.a** Simulation de la commande
6	**II.4.b** Simulateur 3D
12	**III.** Expérimentation

II. Travail réalisé

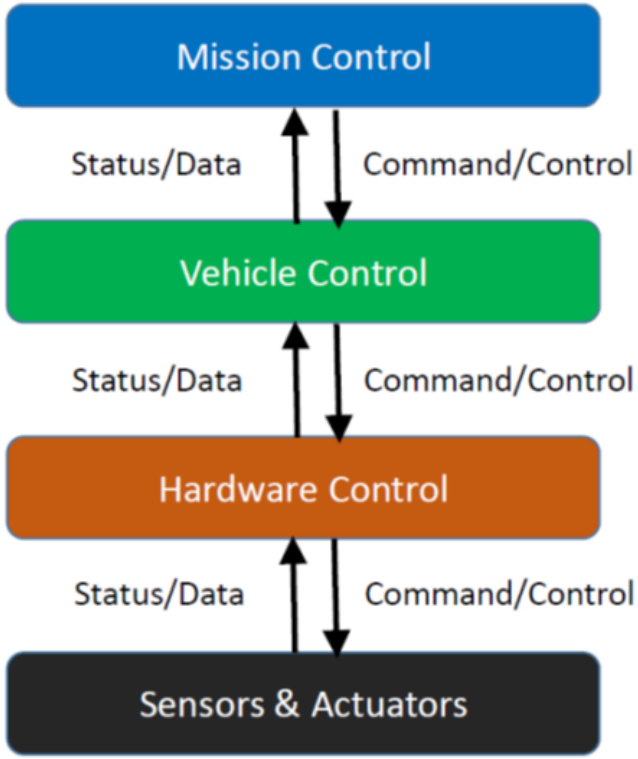
1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycles stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

Riptide MKII µUUV

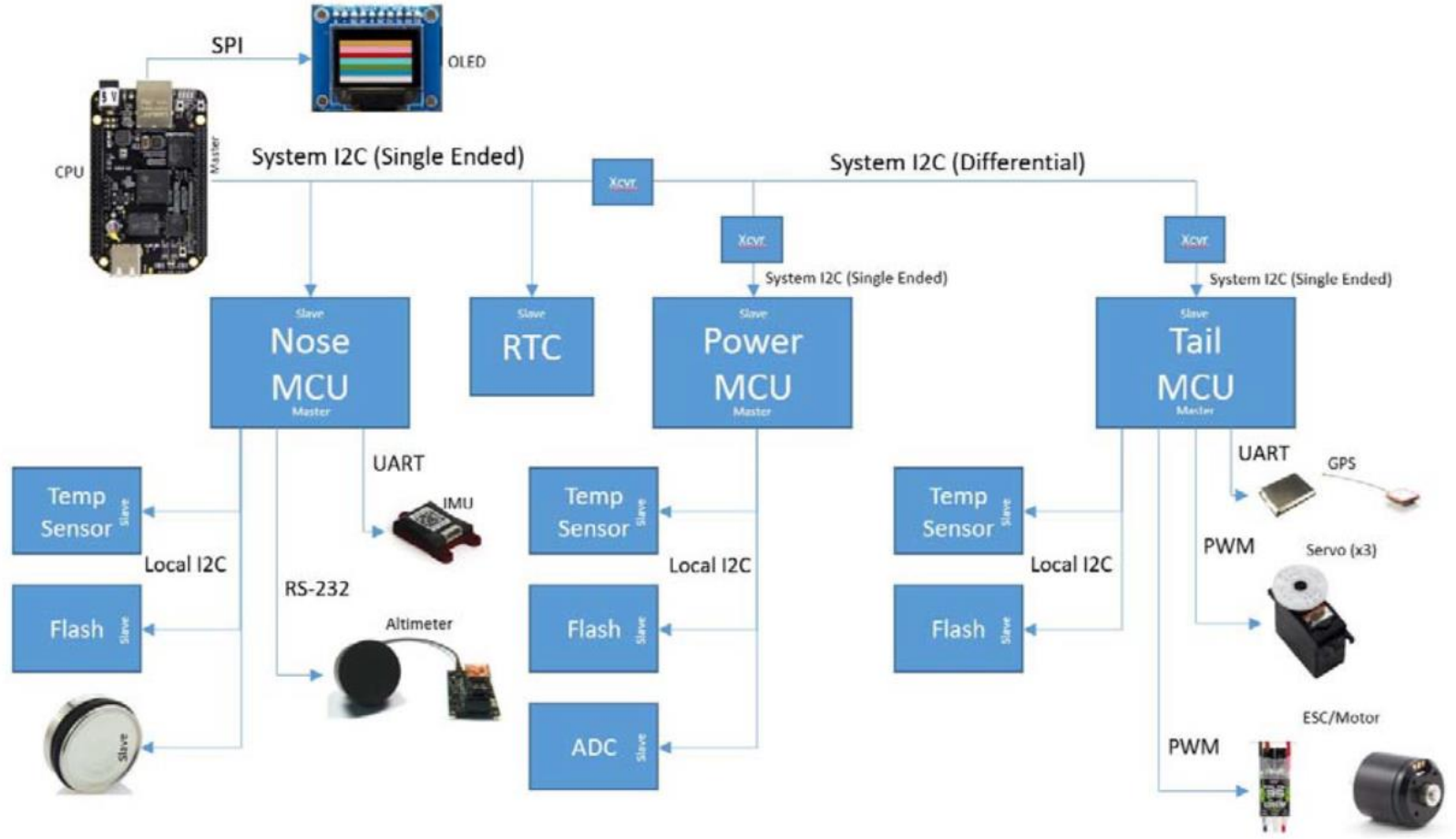
Architecture du Contrôle

•TROIS COUCHES DE CONTRÔLE

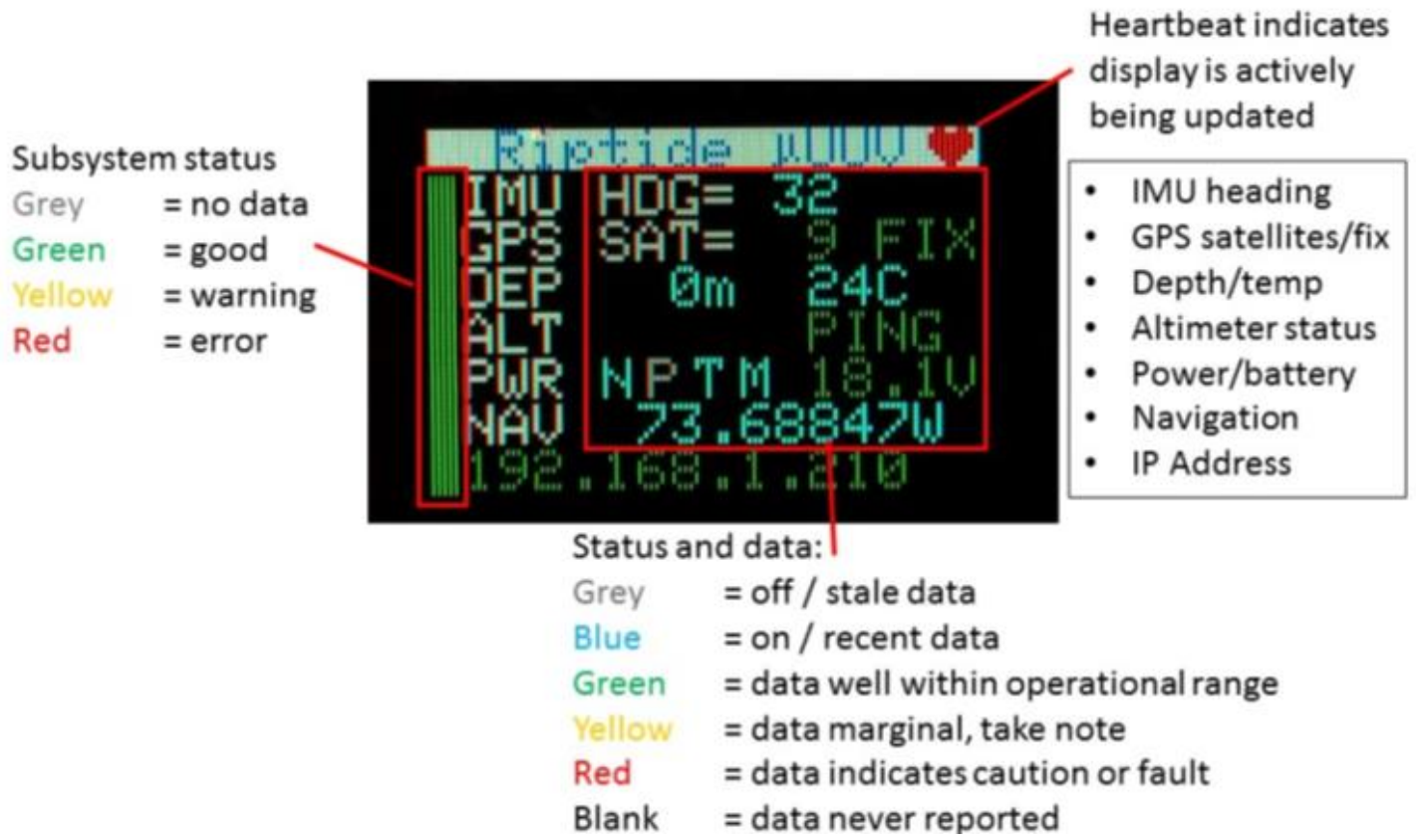
• CONTRÔLE PAR FLUX «DOWN» ET «UP»



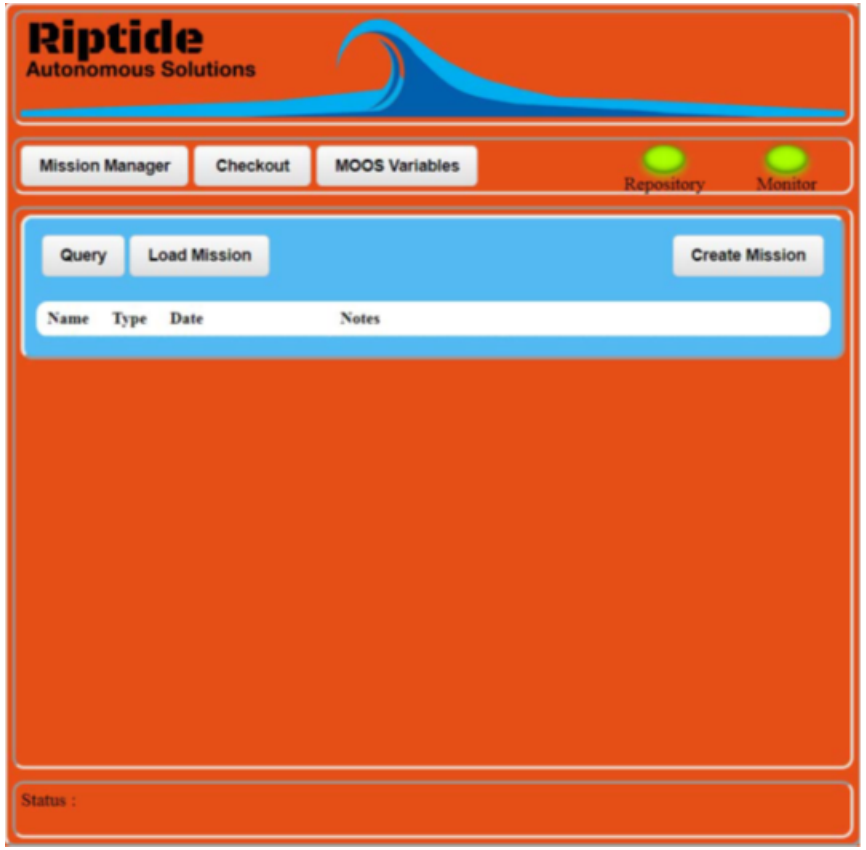
Architecture Electronique



Ecran de Status



Interface Web



Software de départ

- Développement sous MOOS
- Prêt pour des missions sous le bon format
- ROS non installé
- MOOS non installé
- Aucune solution de développement
- Aucun code source ni documentation précise

Passage de MOOS à ROS

Espionner les communications I2C

- Avantage :
  - Pouvoir recoder depuis le départ
  - Enlever MOOS

Mettre un place un bridge en MOOS et ROS

- Avantage
  - Plus simple
  - Les régulateurs en vitesse et en cap sont déjà réglés

- Utilisation des commande ***uPoke*** pour remplacer une valeur de la base de données
- Utilisation des commande ***uXMS*** pour lire des valeurs dans la base de données

Architecture finale

Commande :

- Réception d'une commande par le nœud ROS du bridge
- Écriture dans la base de données sur la variable DESIRED_HEADING
- Processus MOOS (pMarinePID) calcul les commandes moteurs
- Envoi en i2c de la commande moteur

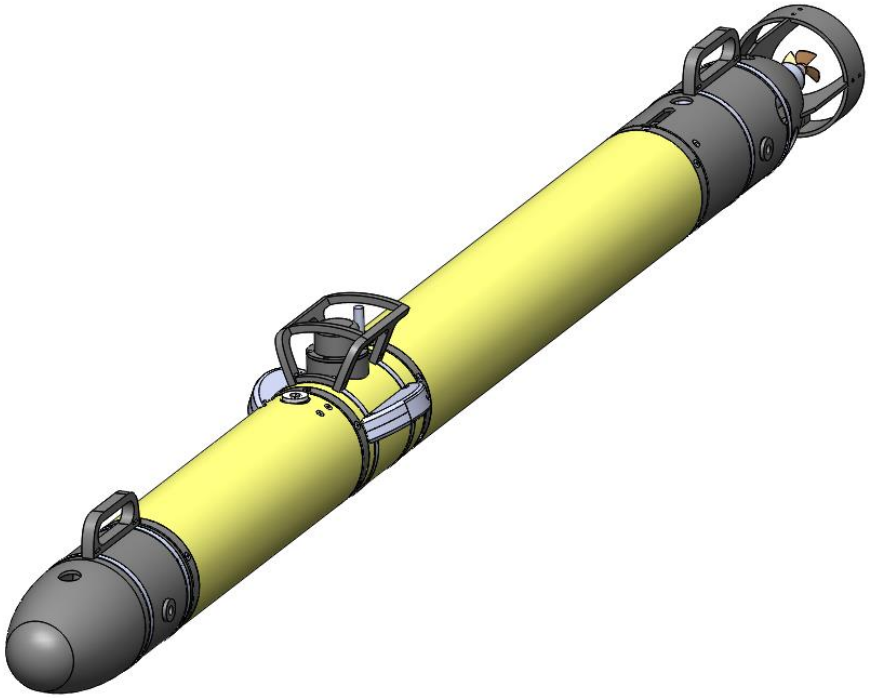
Capteur :

- Mesure des capteurs
- Communication i2c
- Traitement et filtre réalisé par un processus MOOS
- Lecture des valeurs dans la base de données par le nœud ROS du bridge

II. Travail réalisé

1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycles stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

Le Folaga



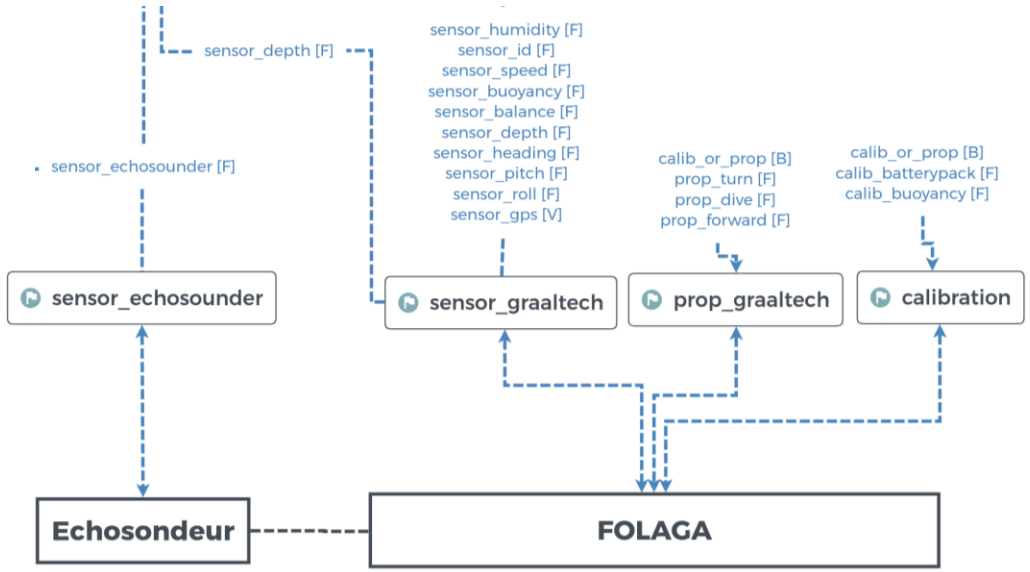
Actionneurs :

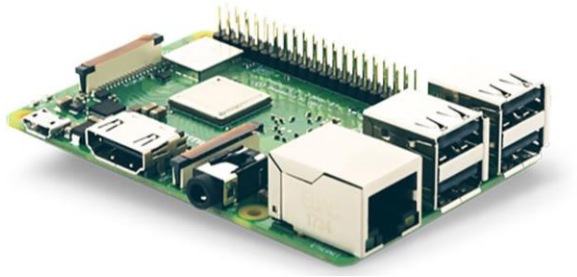
- 8 Pompes pour tourner et plonger
- 1 moteur+hélice pour la propulsion

Capteurs :

- IMU
- GPS
- Capteur de pression

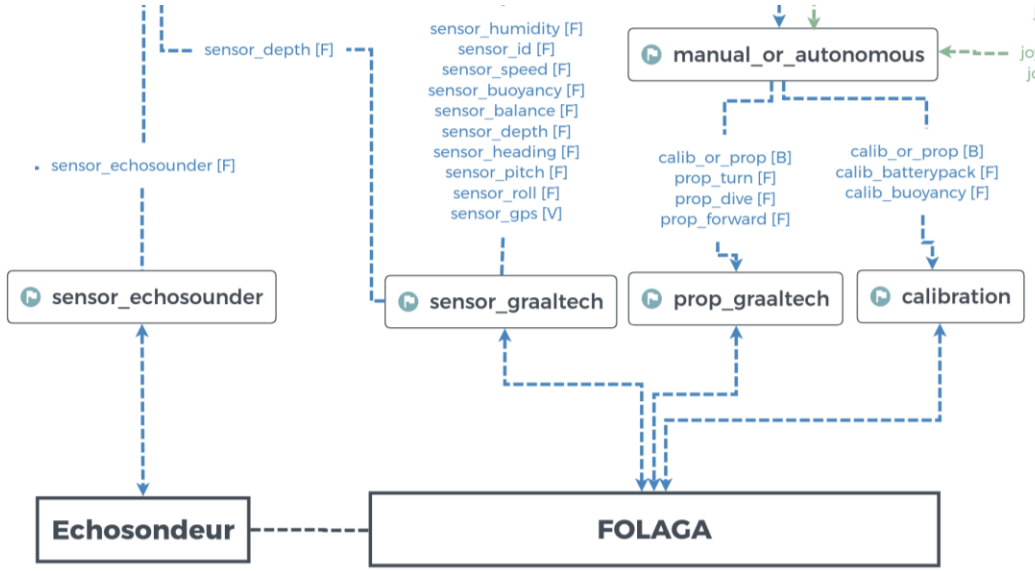
Architecture ROS : Bas niveau



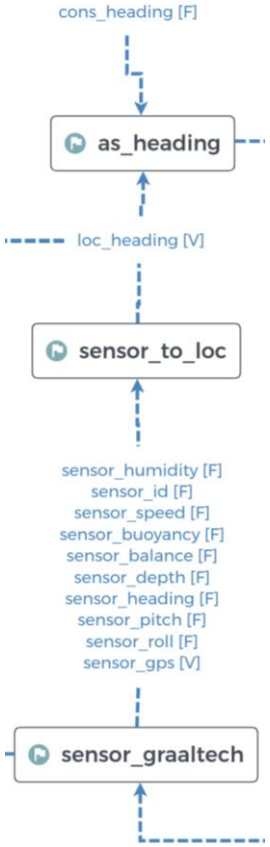


Ordinateur Graaltech

Architecture ROS : mode manuel

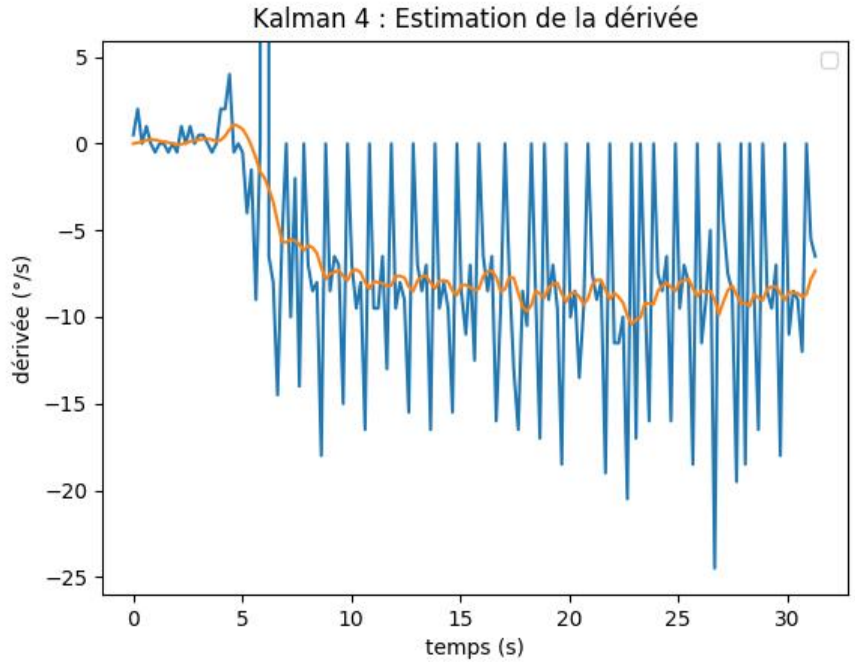


Architecture ROS : Asservissement en cap

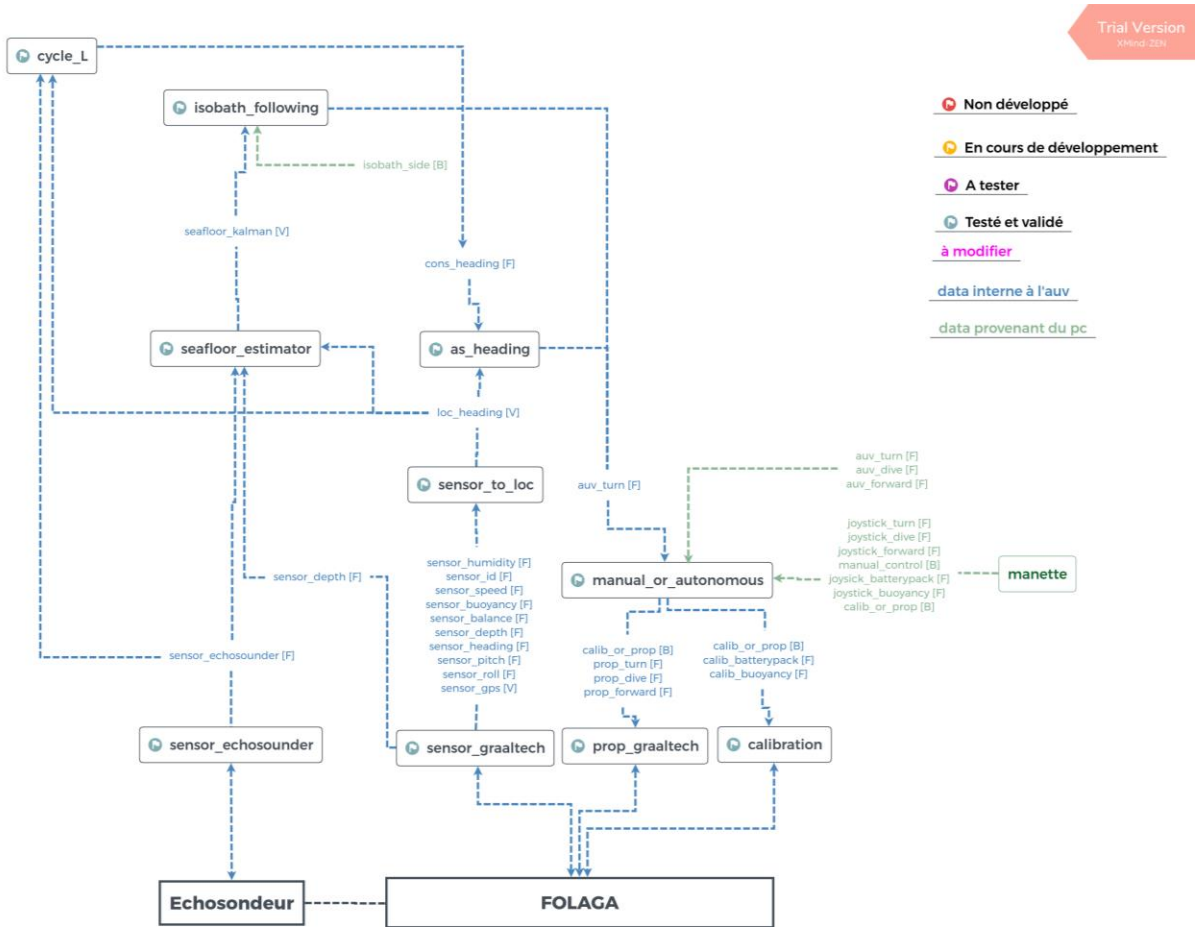


$$x_{k+1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 - dt.x_3.x_2 \\ x_2 + dt.x_3.x_1 \\ x_3 \end{bmatrix}$$

$$y = \begin{bmatrix} \cos(\theta_{mes}) \\ \sin(\theta_{mes}) \\ \frac{st(\theta_{mes} - \theta_{mes-1})}{dt} \\ 1 \end{bmatrix} \sim \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ {x_1}^2 + {x_2}^2 \end{bmatrix}$$



Architecture ROS : Suivi de cycle



II. Travail réalisé

1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycles stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

1. Modèle des AUV

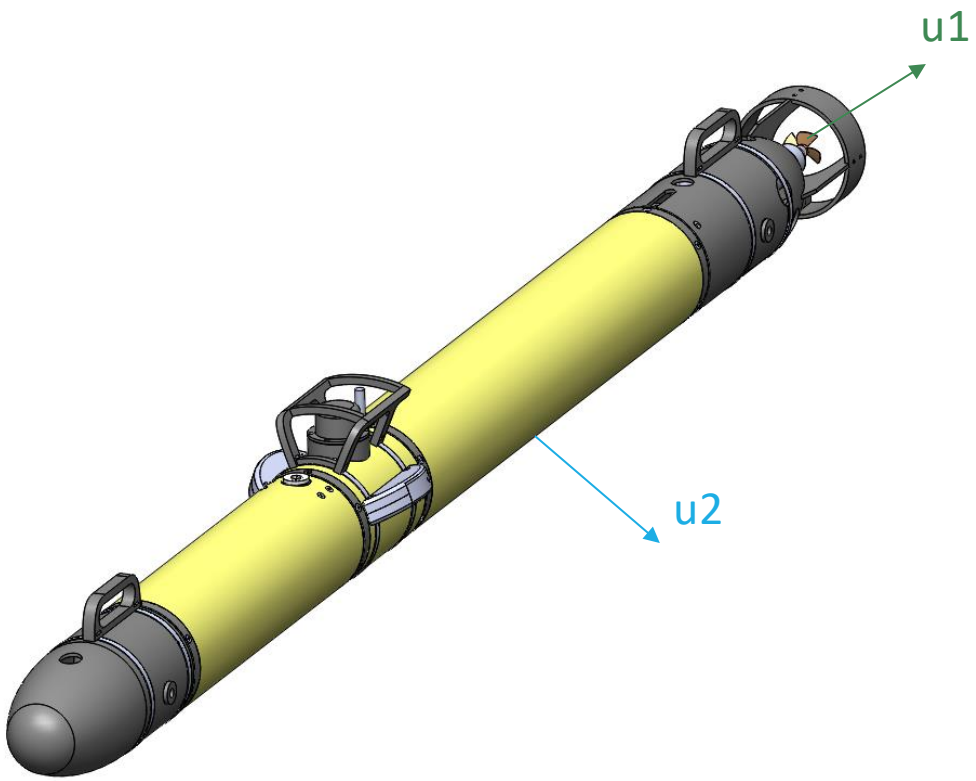
1. Pour le Folaga

Modèle classique

$$\begin{cases} \dot{p} = R(\phi,\theta,\psi) * (v_x, 0, 0)^T \\ \dot{v}_x = p_1 * u_0 * |u_0| - p_2 * v_x * |v_x| \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta).\sin(\phi) & \tan(\theta).\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \dfrac{\sin(\phi)}{\cos(\theta)} & \dfrac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} . \begin{bmatrix} w_{rx} \\ w_{ry} \\ w_{rz} \end{bmatrix} \\ \begin{bmatrix} \dot{w}_{rx} \\ \dot{w}_{ry} \\ \dot{w}_{rz} \end{bmatrix} = (I^{-1}). \begin{bmatrix} 0 \\ 0 \\ p_3 * u_2 * |u_2| - p_4 * w_{rz} * |w_{rz}| \end{bmatrix} \end{cases}$$

Avec hypothèses simplificatrices

$$\begin{cases} \dot{p} = R(\phi,\theta,\psi) * (v_x, 0, 0)^T \\ \dot{v}_x = p_1 * u_0 * |u_0| - p_2 * v_x * |v_x| \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} w_{rx} \\ w_{ry} \\ w_{rz} \end{bmatrix} \\ \begin{bmatrix} \dot{w}_{rx} \\ \dot{w}_{ry} \\ \dot{w}_{rz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ p_5 * u_2 * |u_2| - p_6 * w_{rz} * |w_{rz}| \end{bmatrix} \end{cases}$$

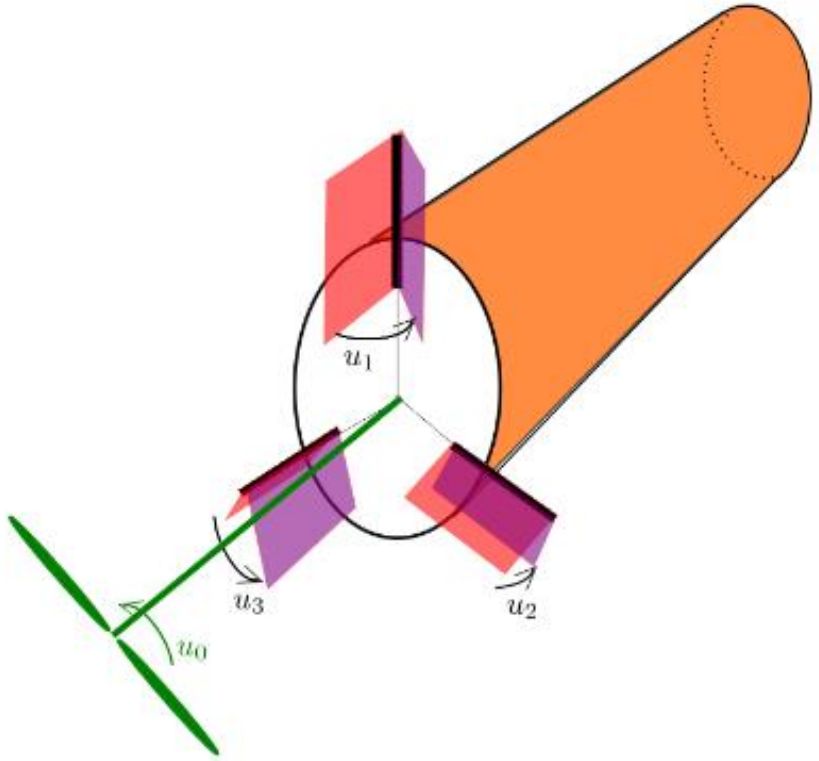


1. Modèle des AUV

2. Pour le Riptide

$$\begin{cases} \dot{p} = R(\phi,\theta,\psi) * (v_x, 0, 0)^T \\ \dot{v}_x = p_1 * u_0 * |u_0| - p_2 * v_x * |v_x| \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta).\sin(\phi) & \tan(\theta).\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} .v_x.\boldsymbol{B}(p3,p4). \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \end{cases}$$

$$\text{avec } B = \begin{pmatrix} p_3 & 0 & 0 \\ 0 & p_4 & 0 \\ 0 & 0 & p_4 \end{pmatrix} . \begin{pmatrix} -1 & -1 & -1 \\ 0 & \sin(2\pi/3) & -\sin(2\pi/3) \\ 1 & \cos(2\pi/3) & \cos(2\pi/3) \end{pmatrix}.$$



II. Travail réalisé

1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycles stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

2. Identification des paramètres

1. Identification des paramètres à rechercher

Pour le Folaga

$$\begin{cases} \dot{p} = R(\phi,\theta,\psi) * (v_x, 0, 0)^T \\ \dot{v}_x = p_1 * u_0 * |u_0| - p_2 * v_x * |v_x| \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} w_{rx} \\ w_{ry} \\ w_{rz} \end{bmatrix} \\ \begin{bmatrix} \dot{w}_{rx} \\ \dot{w}_{ry} \\ \dot{w}_{rz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ p_5 * u_2 * |u_2| - p_6 * w_{rz} * |w_{rz}| \end{bmatrix} \end{cases}$$

Pour le Riptide

$$\begin{cases} \dot{p} = R(\phi,\theta,\psi) * (v_x, 0, 0)^T \\ \dot{v}_x = p_1 * u_0 * |u_0| - p_2 * v_x * |v_x| \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta).\sin(\phi) & \tan(\theta).\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} .v_x.\boldsymbol{B}(p3,p4). \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \end{cases}$$

$$\text{avec } B = \begin{pmatrix} p_3 & 0 & 0 \\ 0 & p_4 & 0 \\ 0 & 0 & p_4 \end{pmatrix} . \begin{pmatrix} -1 & -1 & -1 \\ 0 & \sin(2\pi/3) & -\sin(2\pi/3) \\ 1 & \cos(2\pi/3) & \cos(2\pi/3) \end{pmatrix}.$$

Approximation via la matrice d'inertie d'un cylindre

2. Identification des paramètres

2. Protocole d'obtention des paramètres dynamiques

Protocole :
1 : commande constante, atteinte de la vitesse Vcst
2 : arrêt de la commande

En 2 : $-p_{frot}.V.|V| = a = 0$ $V(t) = \frac{V_{cst}}{V_{cst}.p_{frot}.t + 1}$ $M(t) = M(0) + \frac{log(V_{cte}.p_{frot}.t + 1)}{p_{frot}}$

En 1 : $p_{prop} * u * |u| - p_{frot} * V * |V| = 0$

2. Identification des paramètres

2. Protocole d'obtention des paramètres dynamiques par la méthode des intervalles

Méthodes :
1 : commande constante, puis arrêt de la commande
2 : Modèle complet

En 1 :

$$\psi(t) = \psi(0) + \frac{\log(\dot{\psi}(0) * p6 * t + 1)}{p6} \qquad \ddot{\psi} = p5 * u3 * |u3| - p6 * \dot{\psi} * |\dot{\psi}|$$

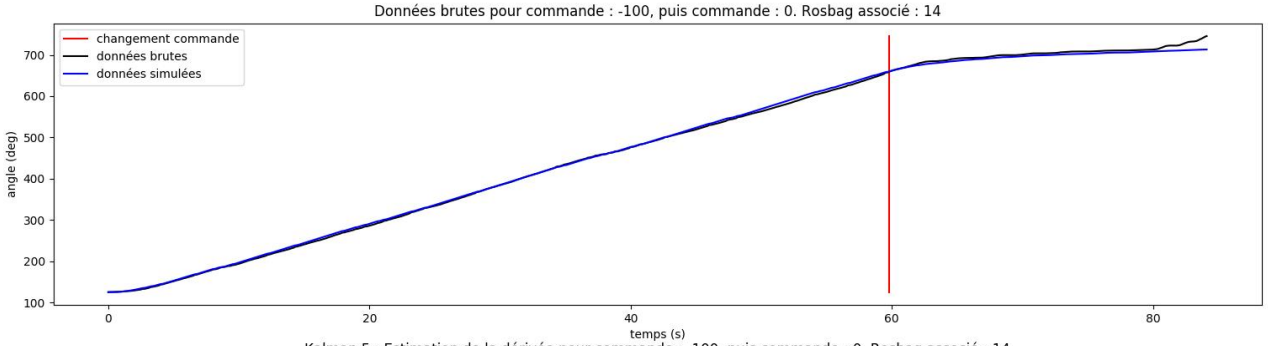
En 2 :

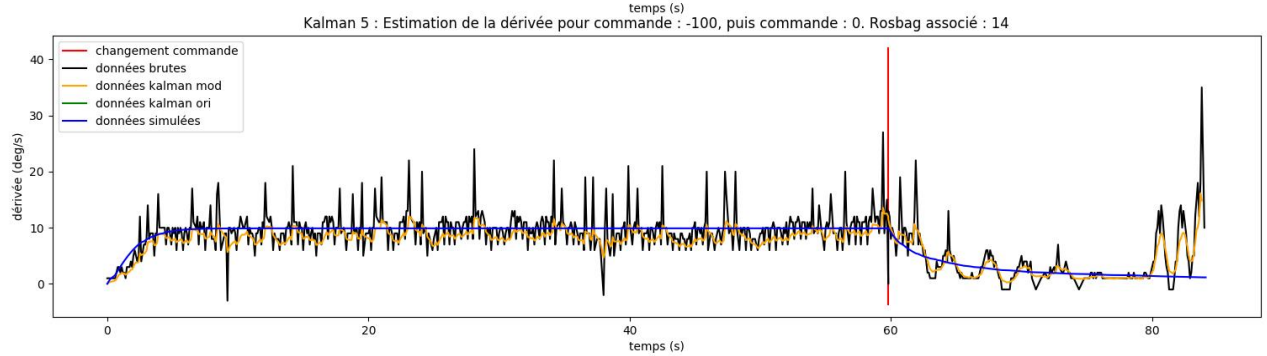
$$\ddot{\psi} = p5 * u3 * |u3| - p6 * \dot{\psi} * |\dot{\psi}|$$

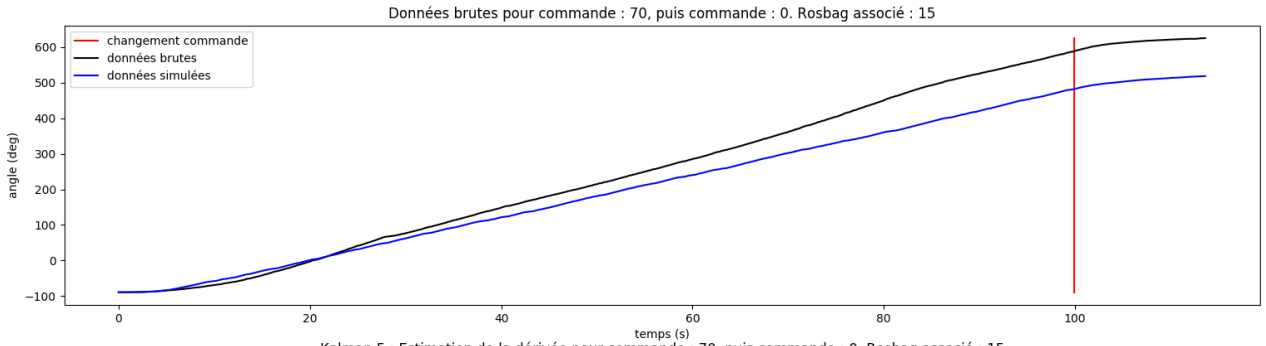
2. Identification des paramètres

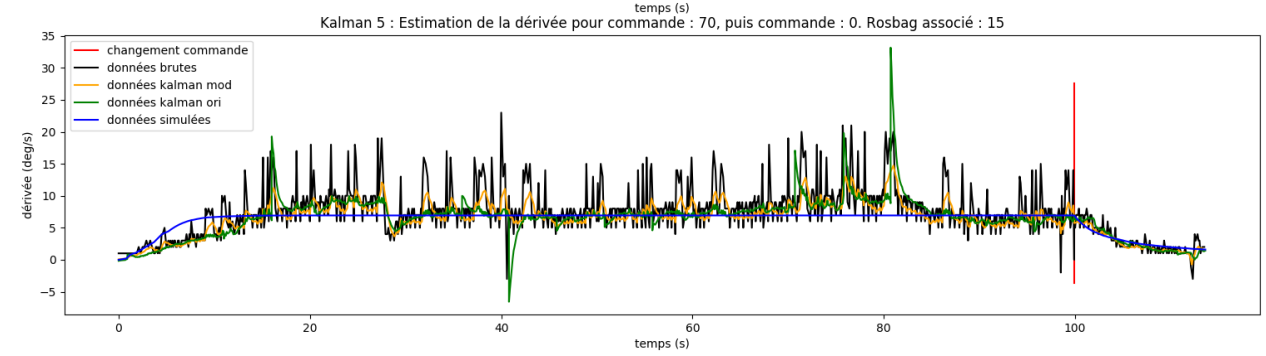
3. Application pour les coefficients en rotation du Folaga

Obtention via minimisation par moindres carrés : mauvais résultats au lac Ty Colo et dans la piscine de l'ENSTA, bons résultats dans la Penfeld.



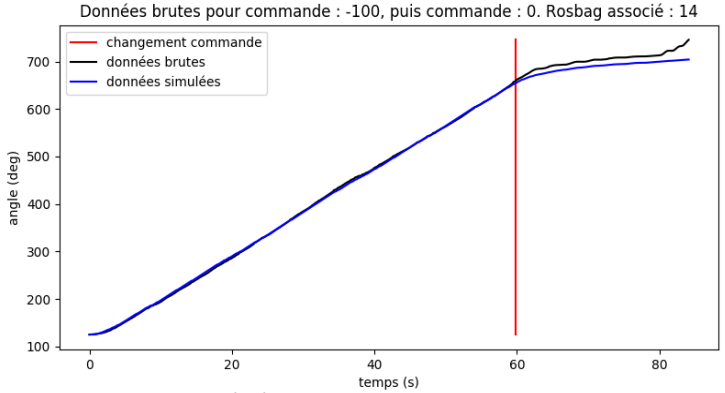


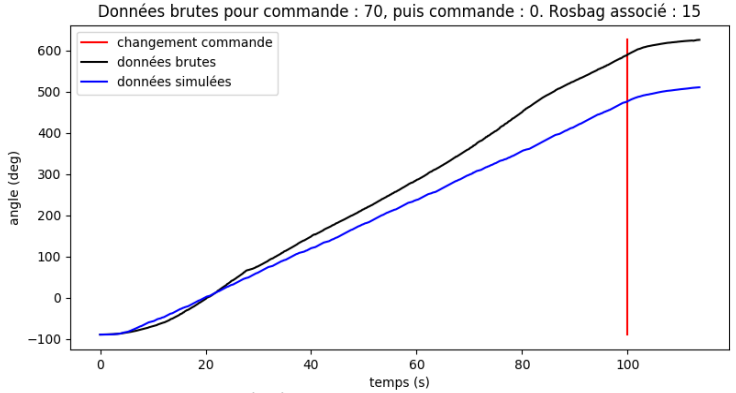


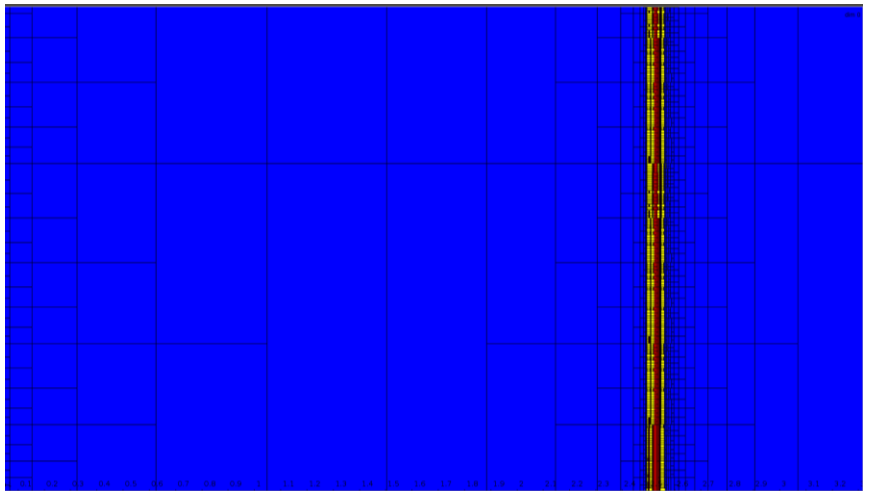


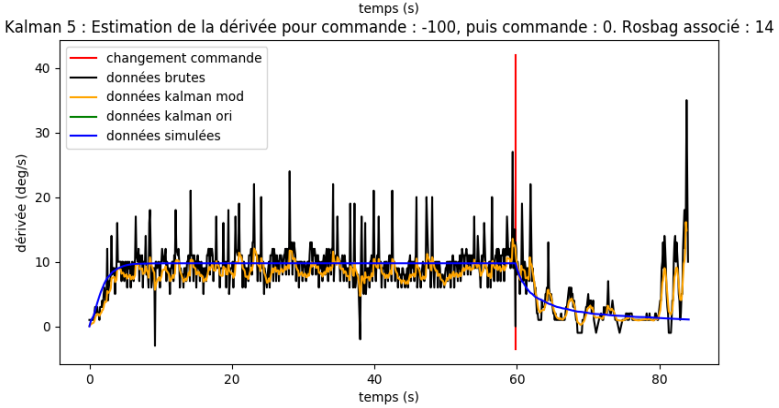
2. Identification des paramètres

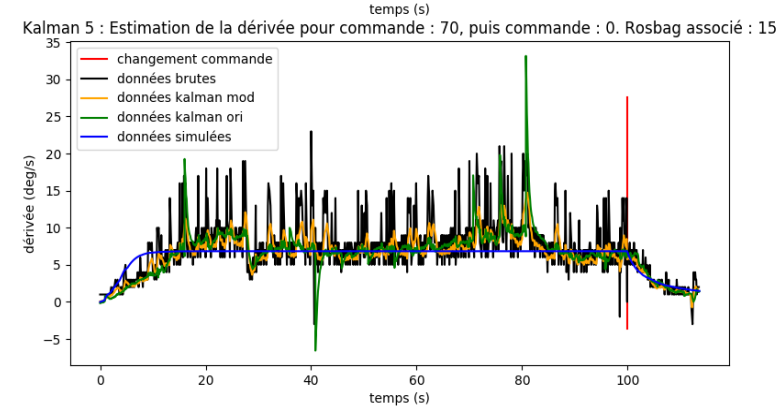
3. Application pour les coefficients en rotation du Folaga par la méthode des intervalles







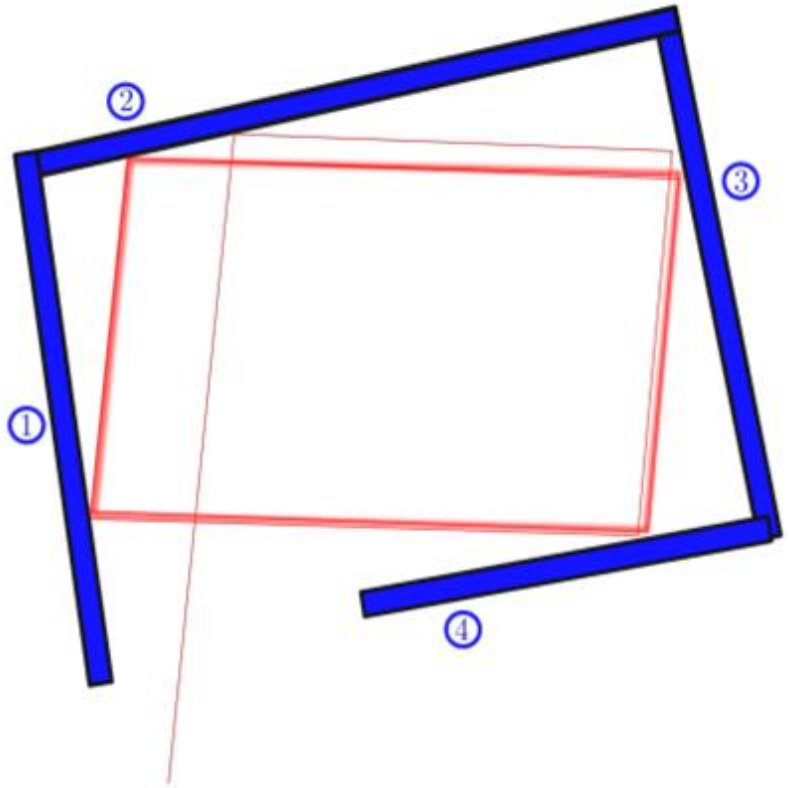


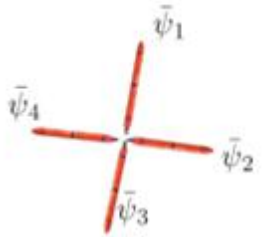


II. Travail réalisé

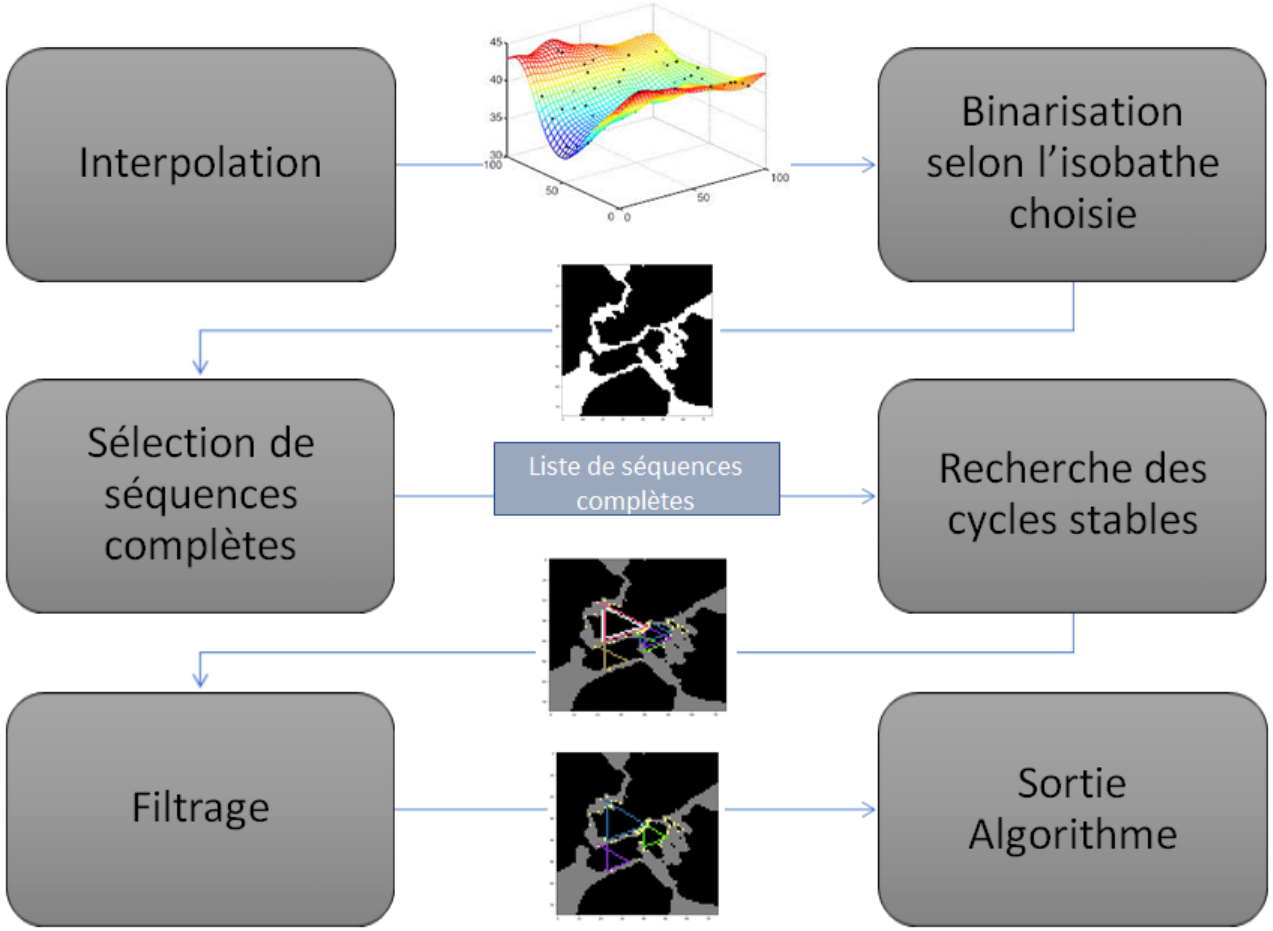
1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycles stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

Définition d'un cycle stable

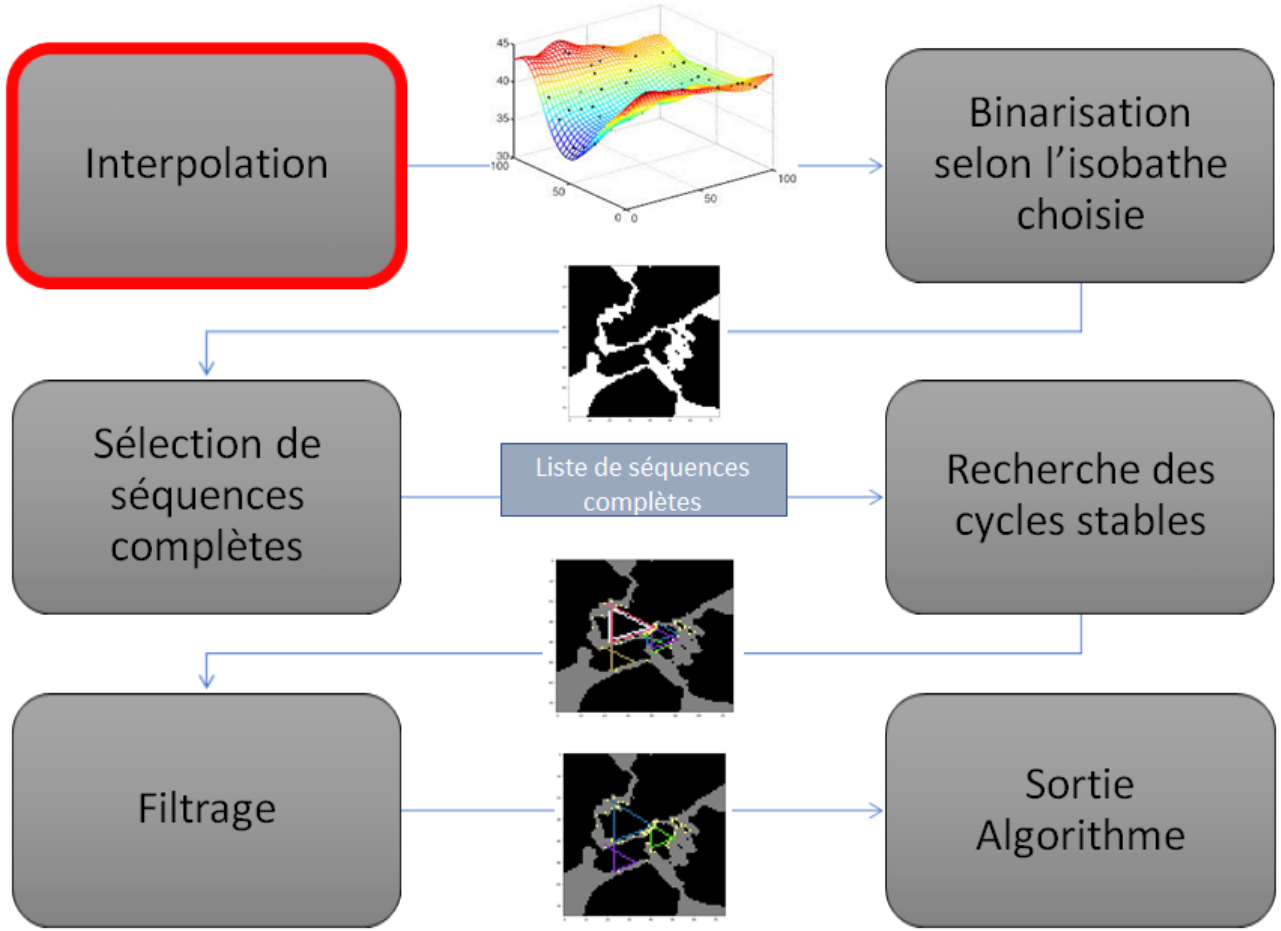




Etapes de l'algorithme

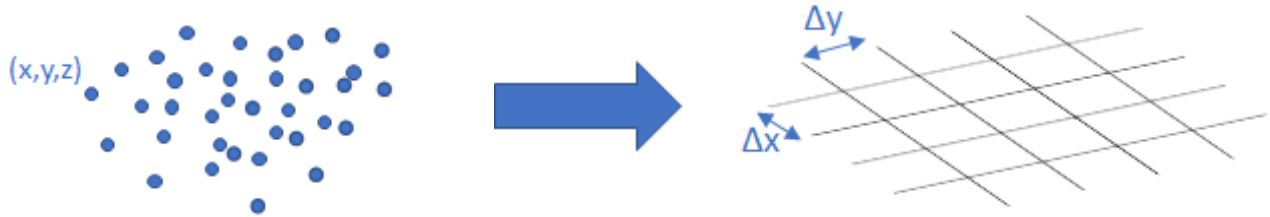


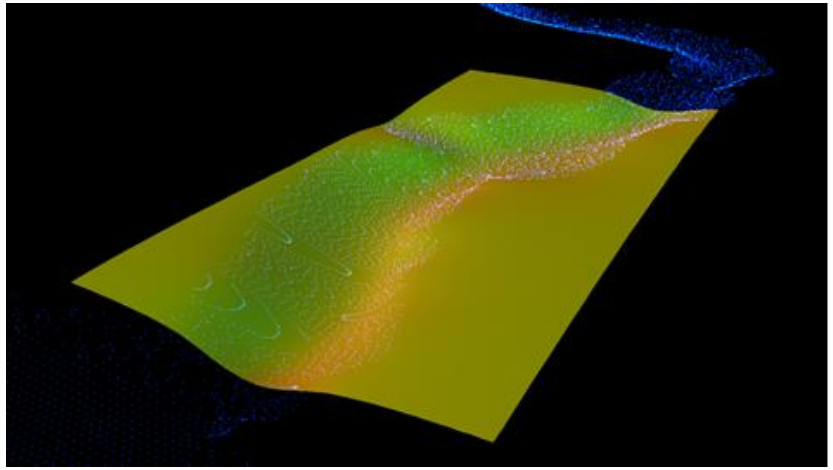
Etapes de l'algorithme

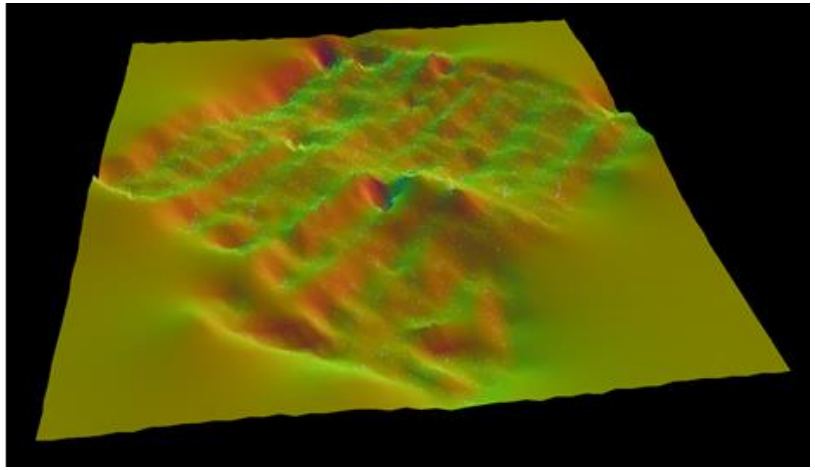


Interpolation

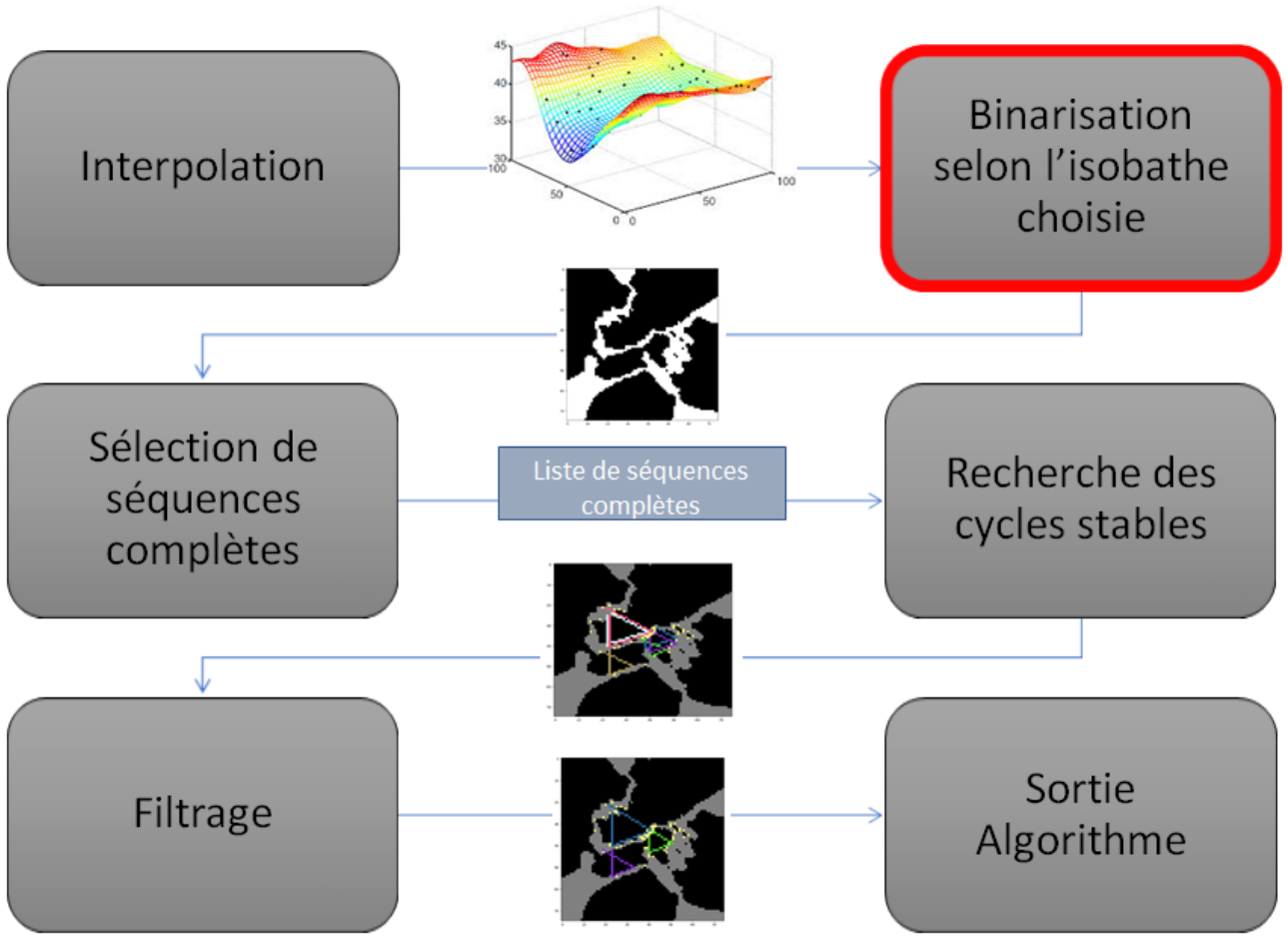
- Krigeage ordinaire:
  - Minimisation de la variance de l'erreur de prévision
  - Nécessite un modèle de dépendance spatiale





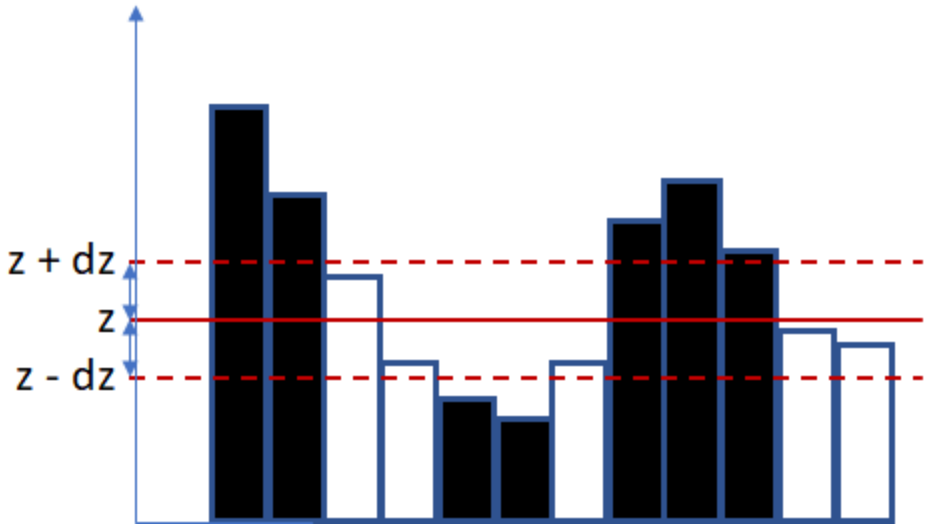


Etapes de l'algorithme

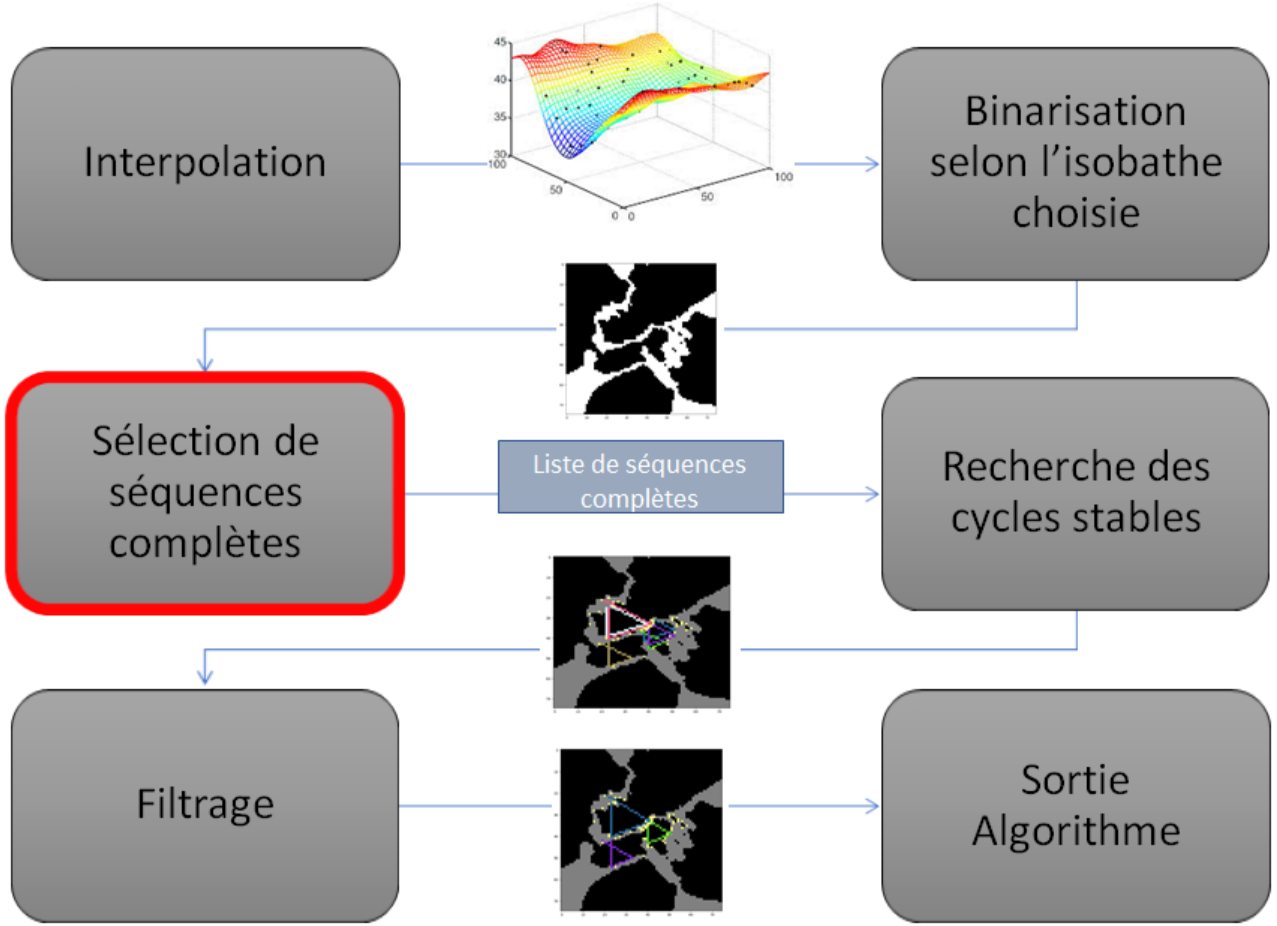


Binarisation

- Choix d'une profondeur z
- Choix d'une marge de détection dz

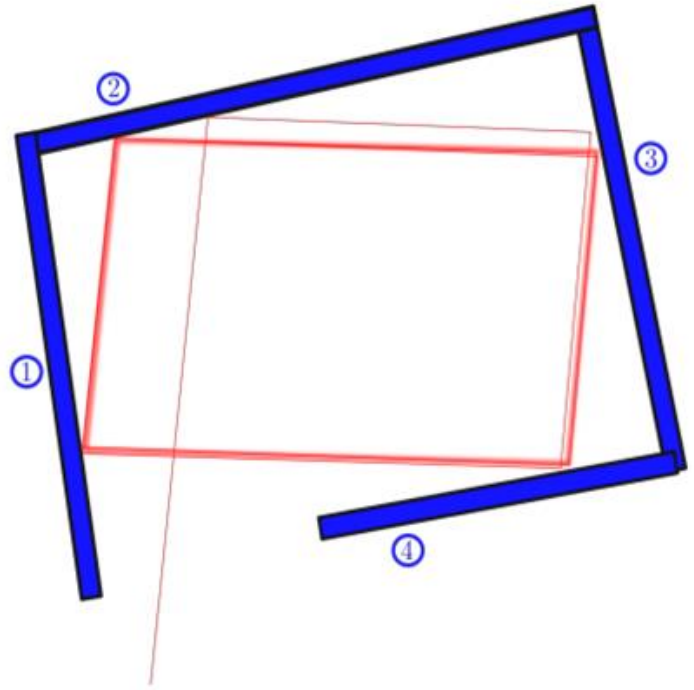


Etapes de l'algorithme



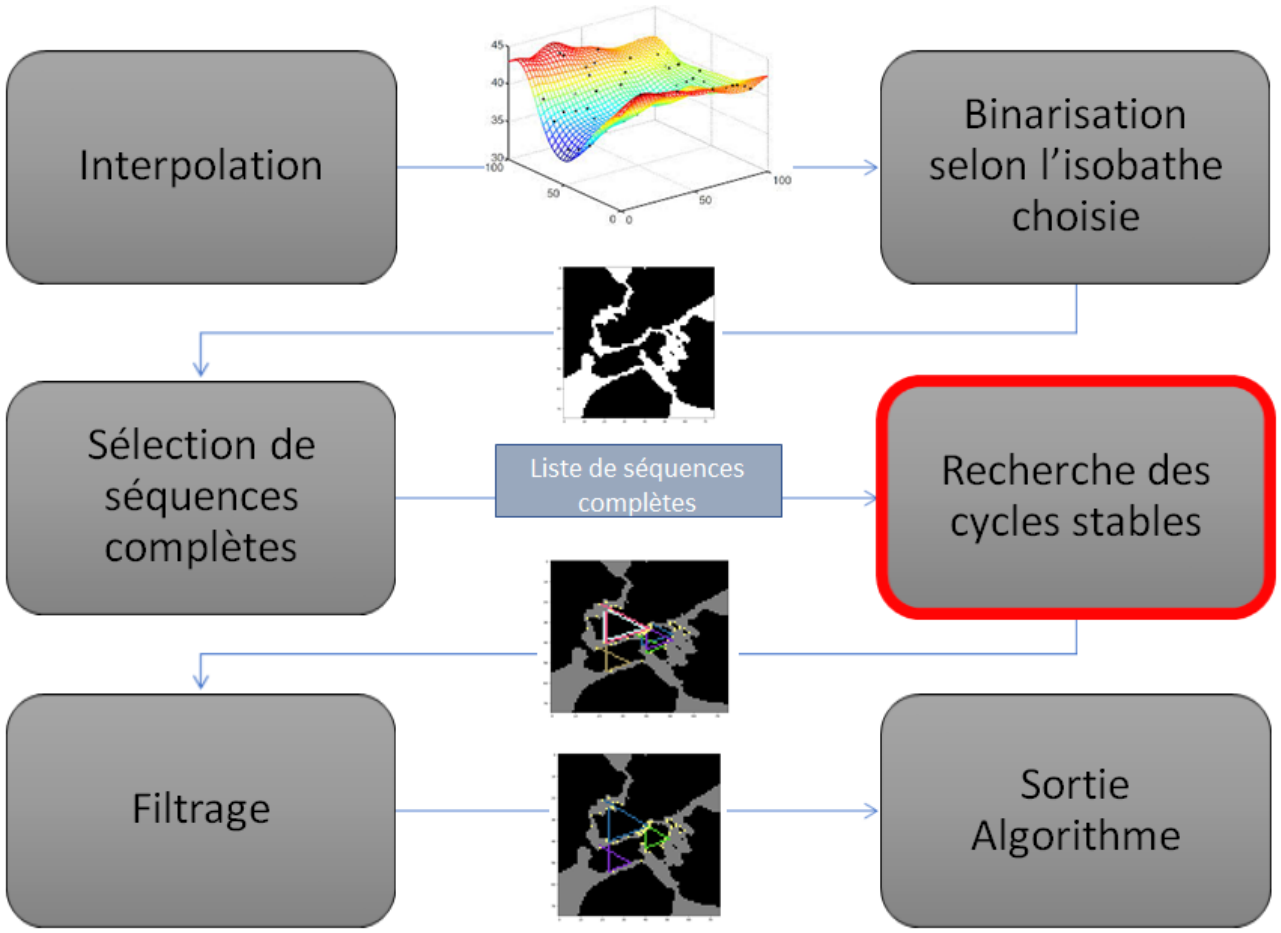
Sélection des séquences complètes

- Critère: Nombre de rebonds



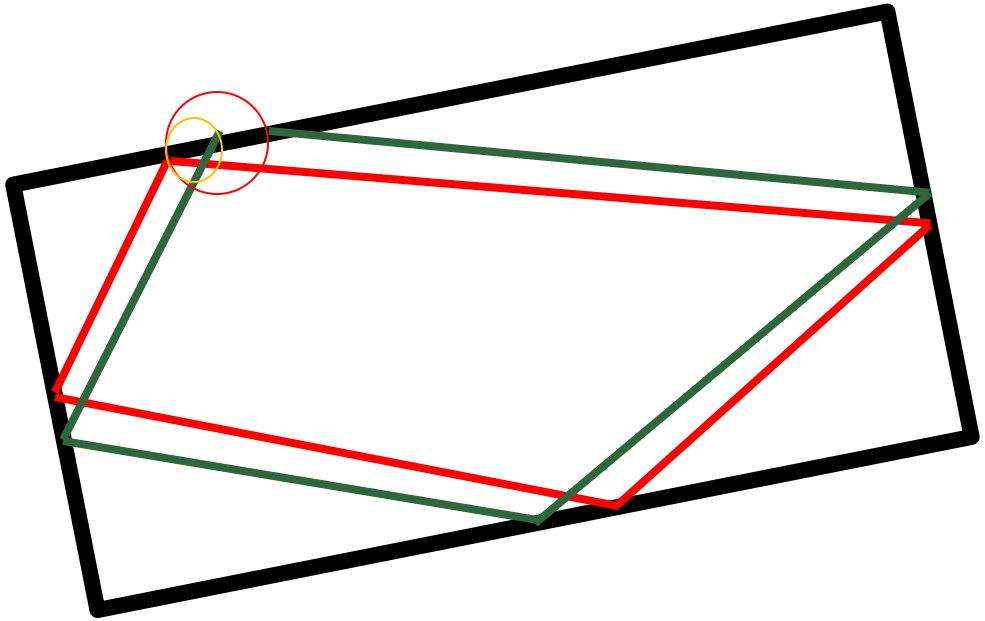
séquence complète = ① + ② + ③ + ④

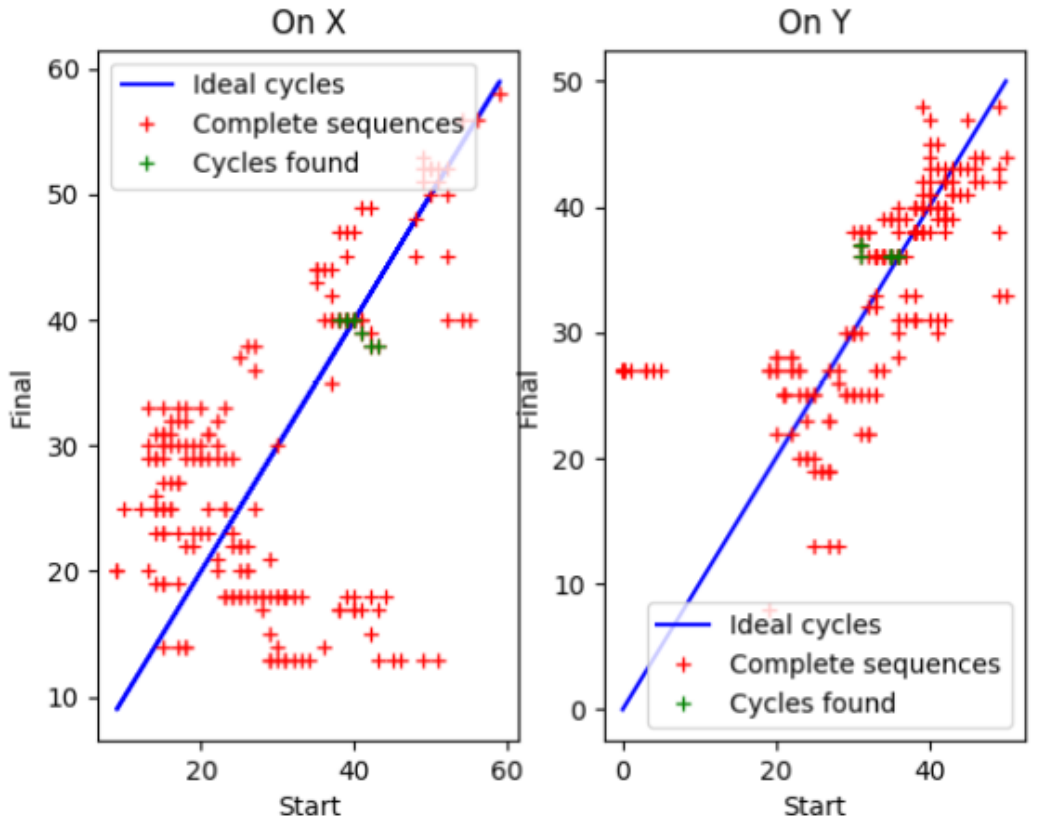
Etapes de l'algorithme



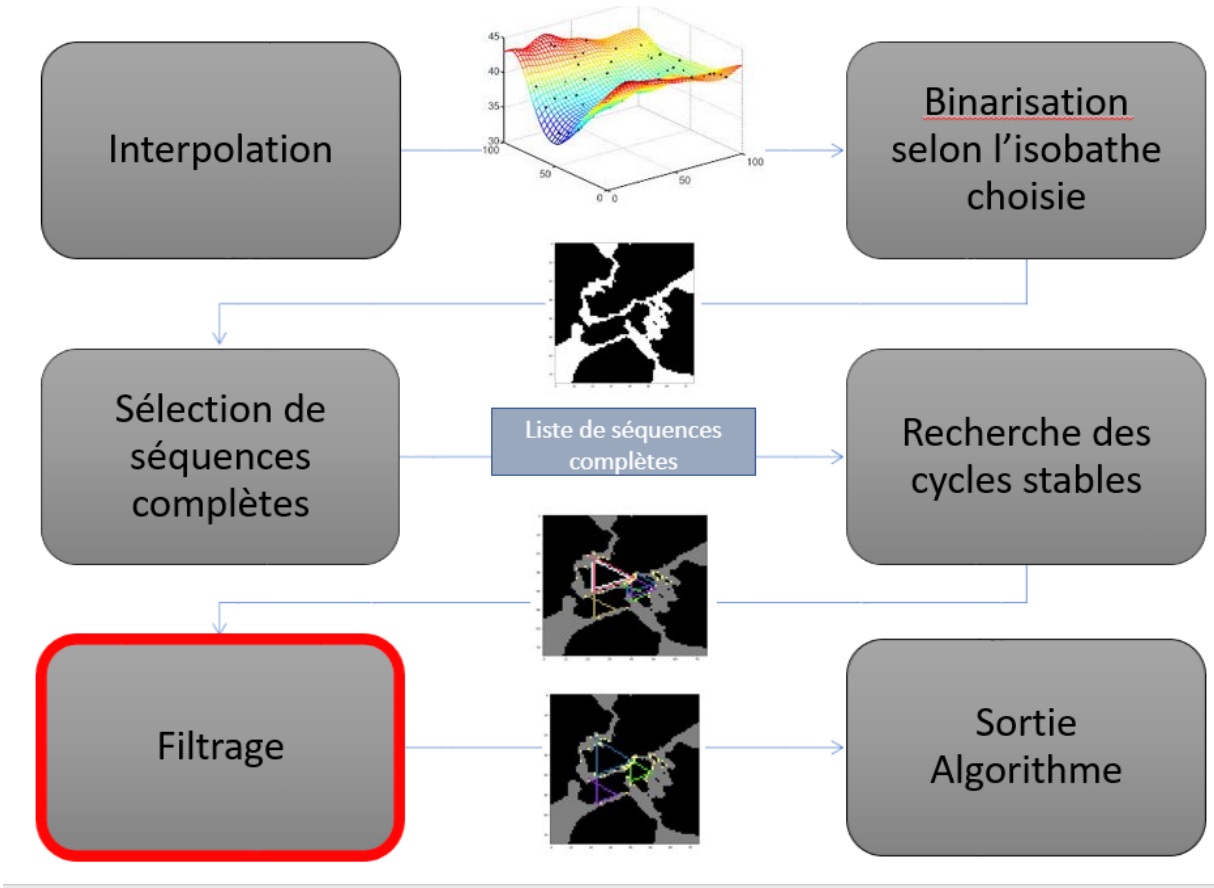
Recherche des cycles stables

- Critère de Banach

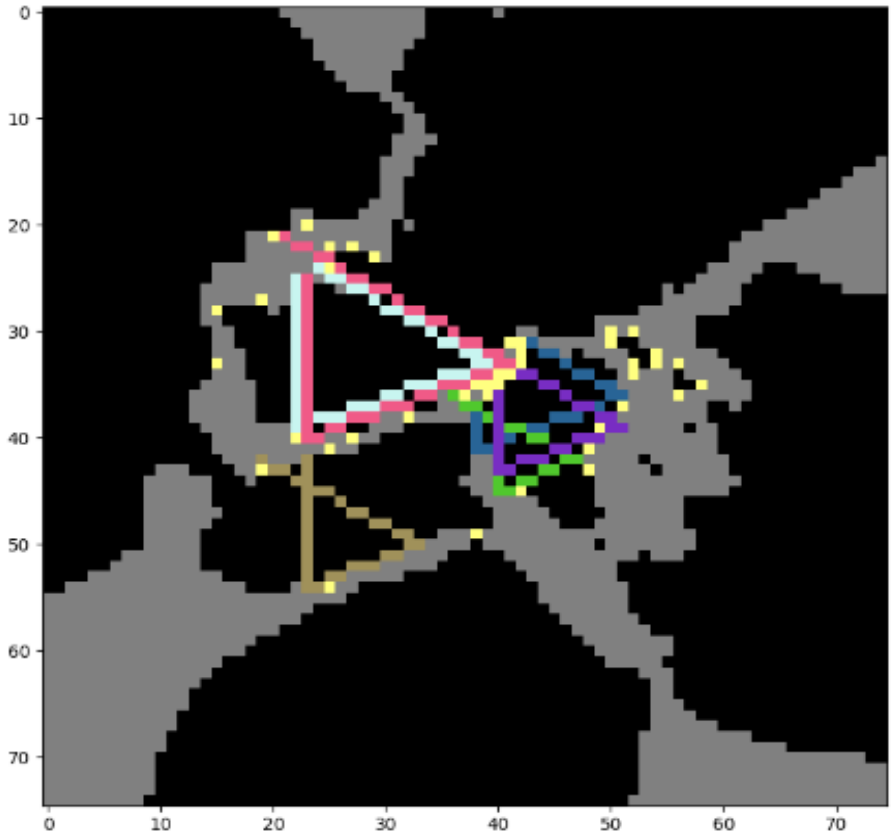


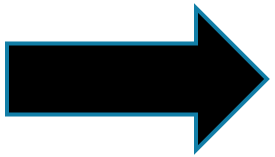


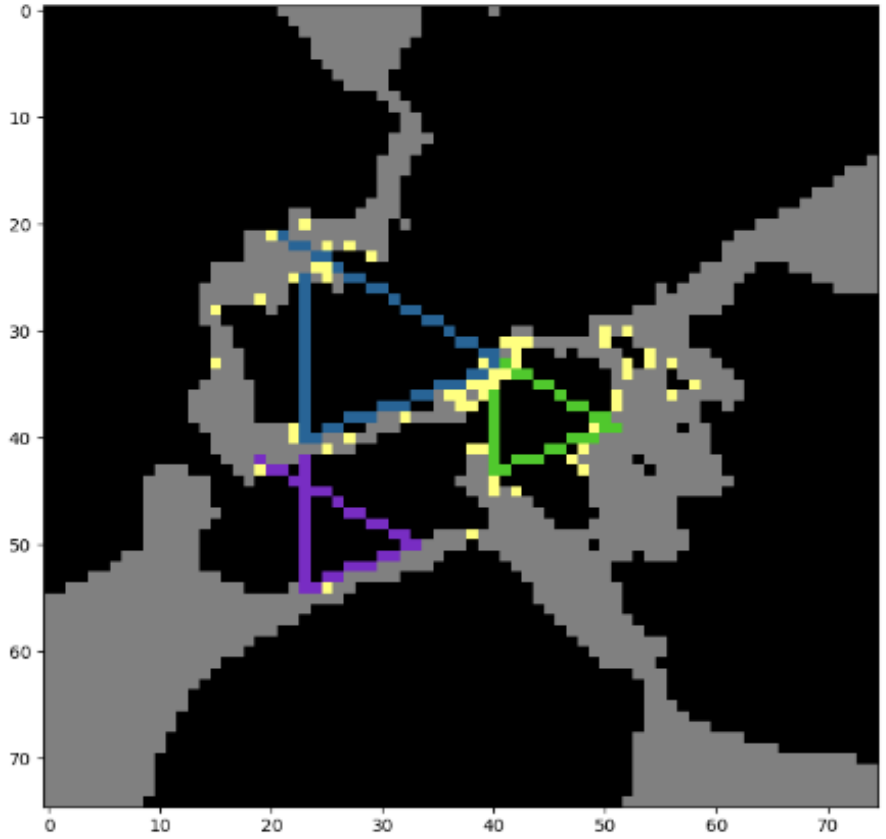
Etapes de l'algorithme



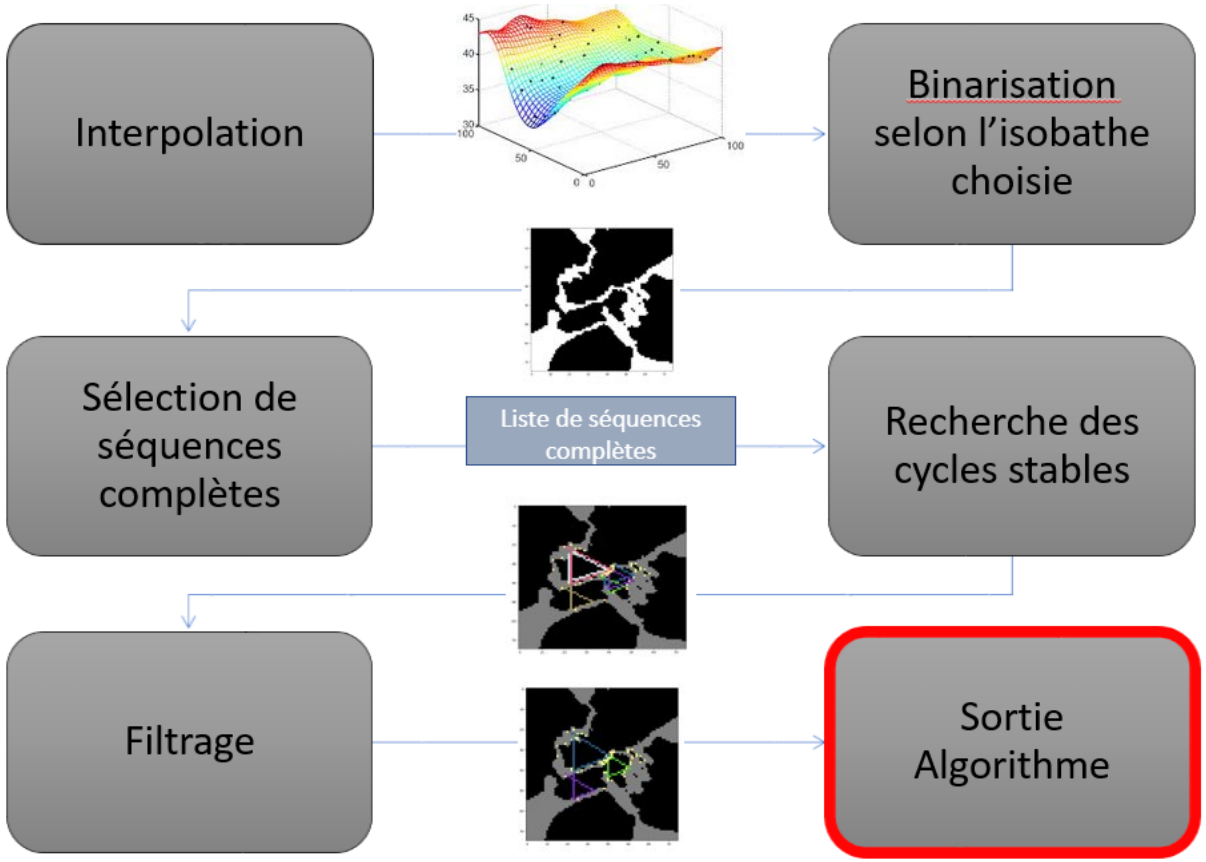
Filtrage


0
10
20
30
40
50
60
70
0
10
20
30
40
50
60
70




0
10
20
30
40
50
60
70
0
10
20
30
40
50
60
70

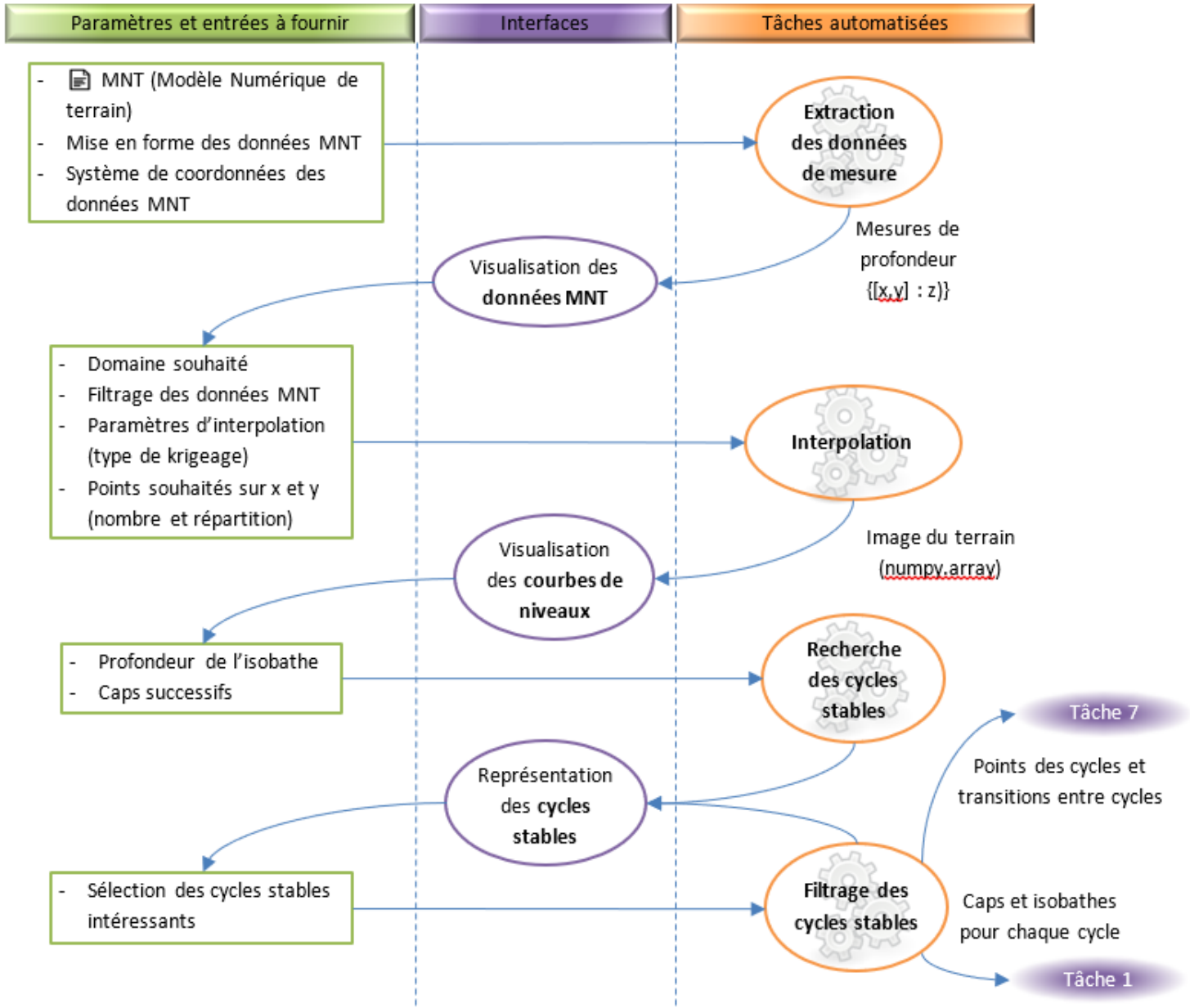
Etapes de l'algorithme



Sortie de l'algorithme

```
Searching stable cycles...
[[[21 21]
  [33 41]
  [40 23]
  [25 23]]

 [[24 24]
  [33 39]
  [39 22]
  [25 22]]

 [[31 43]
  [36 51]
  [41 38]
  [37 38]]

 [[33 41]
  [39 51]
  [43 40]
  [36 40]]

 [[36 36]
  [42 47]
  [45 40]
  [36 40]]

 [[42 19]
  [50 33]
  [54 23]
  [42 23]]]
There are 6 zones
There are respectively composed by [3, 2, 7, 3, 11, 1]_cycles
```

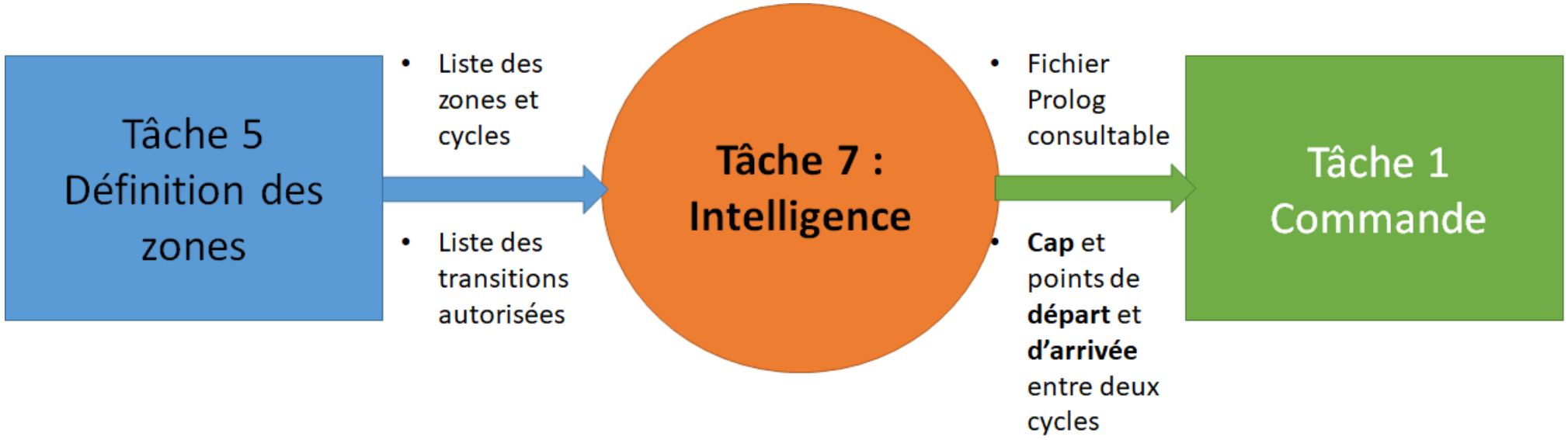
Paramètres


Paramètres et entrées à fournir
Interfaces
Tâches automatisées
- MNT (Modèle Numérique de terrain)
- Mise en forme des données MNT
- Système de coordonnées des données MNT
Extraction des données de mesure
Mesures de profondeur {[x,y] : z)}
Visualisation des données MNT
- Domaine souhaité
- Filtrage des données MNT
- Paramètres d'interpolation (type de krigeage)
- Points souhaités sur x et y (nombre et répartition)
Interpolation
Image du terrain (numpy.array)
Visualisation des courbes de niveaux
- Profondeur de l'isobathe
- Caps successifs
Recherche des cycles stables
Tâche 7
Points des cycles et transitions entre cycles
Représentation des cycles stables
- Sélection des cycles stables intéressants
Filtrage des cycles stables
Caps et isobathes pour chaque cycle
Tâche 1

II. Travail réalisé

1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycles stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

Position de la tâche dans le projet



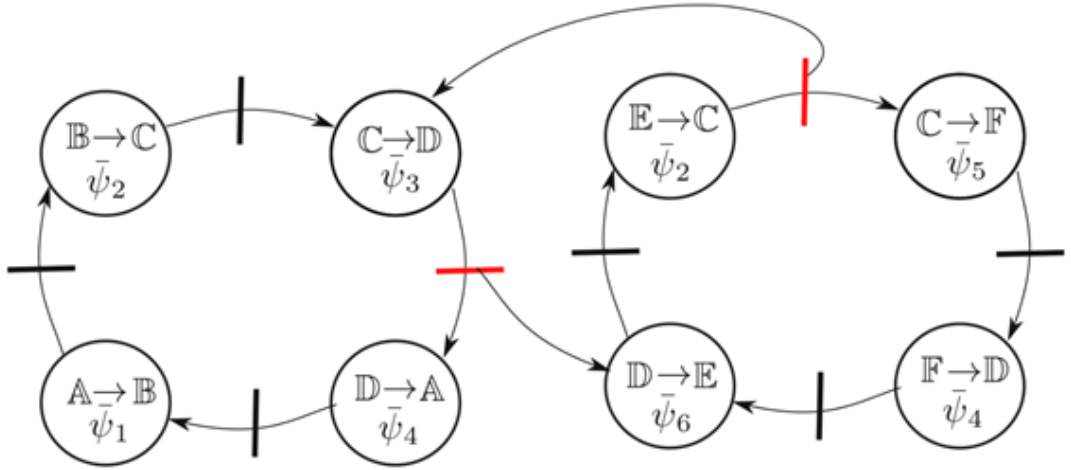
Travail effectué

- Travail en deux temps :
  - Création d'un fichier Prolog sur un exemple particulier
  - Génération automatique de ce fichier pour un ensemble de zones quelconques

- Le fichier final est donc adapté aux sorties de la tâche 5

Travail effectué

- Trois étapes pour la génération du code Prolog :
  1. Générer les états
  2. Générer les transitions
  3. Développer un algorithme de recherche de chemin



Représentation schématique

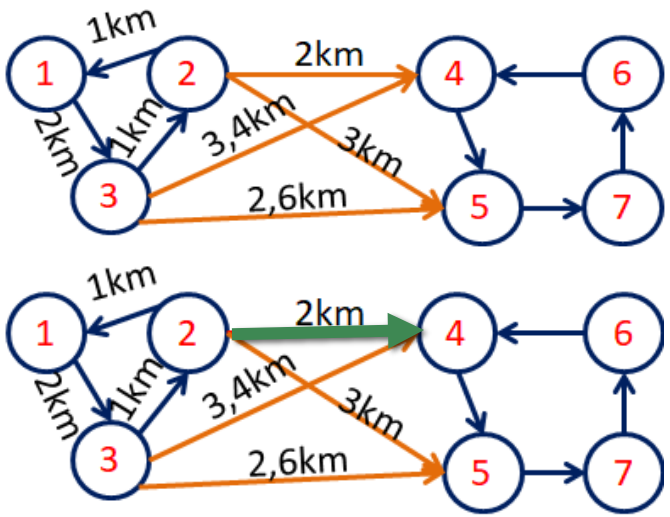
1. **Récupération** des cycles et transitions

2. **Génération automatique** du code prolog

3. Calcul du **plus court chemin** entre deux cycles

4. **Renvoi** du cap et des 2 points extrêmes

- 1) [ZonesCycle 1, ZonesCycle 2…, ZonesCycle n]
- 2) [Transitions 1 à 2, Transitions 2 à 3 , … Transitions n-1 à n



Requête : Cycle 1 à Cycle 2
Réponse : Zone 2, Zone 4, Cap(Zone2,Zone4)

Travail effectué

- Algorithme de recherche de chemin en Prolog:
  - Prise en compte des distances entre les points automatiquement calculés
  - Liste de tous les chemins possibles entre deux zones, et calcul du plus court
- Traitement du chemin généré sous Python:
  - Détermination des changements de cycles
  - En retourner les caps et points extrêmes

Evolutions possibles

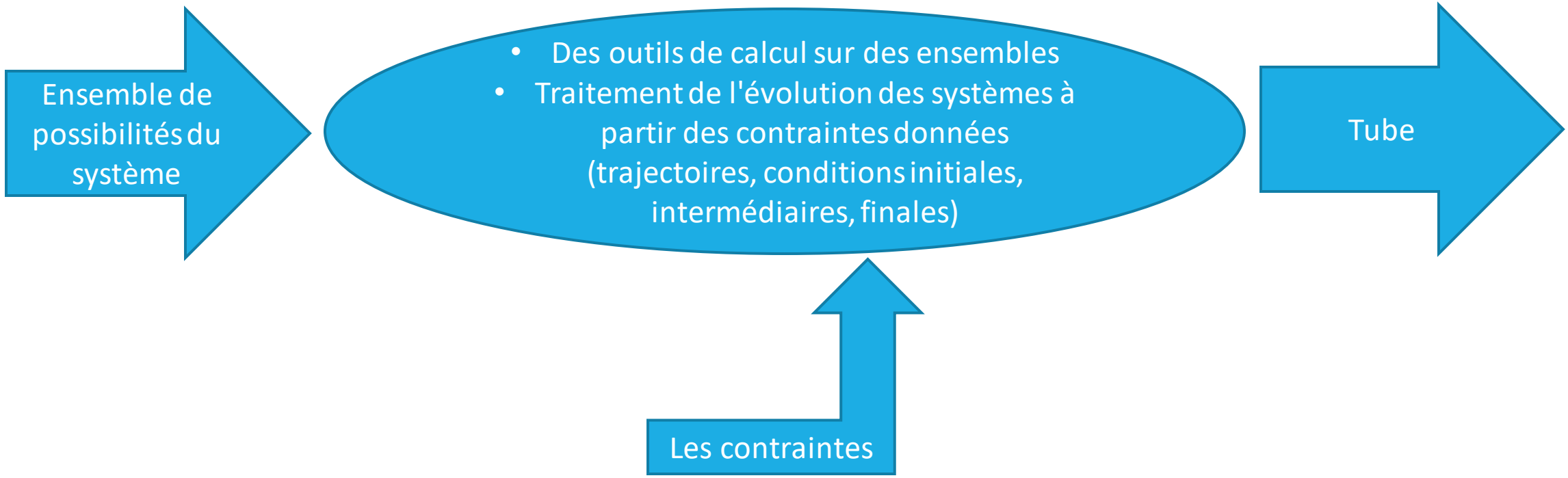
- Gestion des requêtes inattendues comme :
  - L'évitement de zones (ou obstacles)
  - Le passage obligé par une zone

- Générer un automate complet qui gère la commande de l'AUV en cap

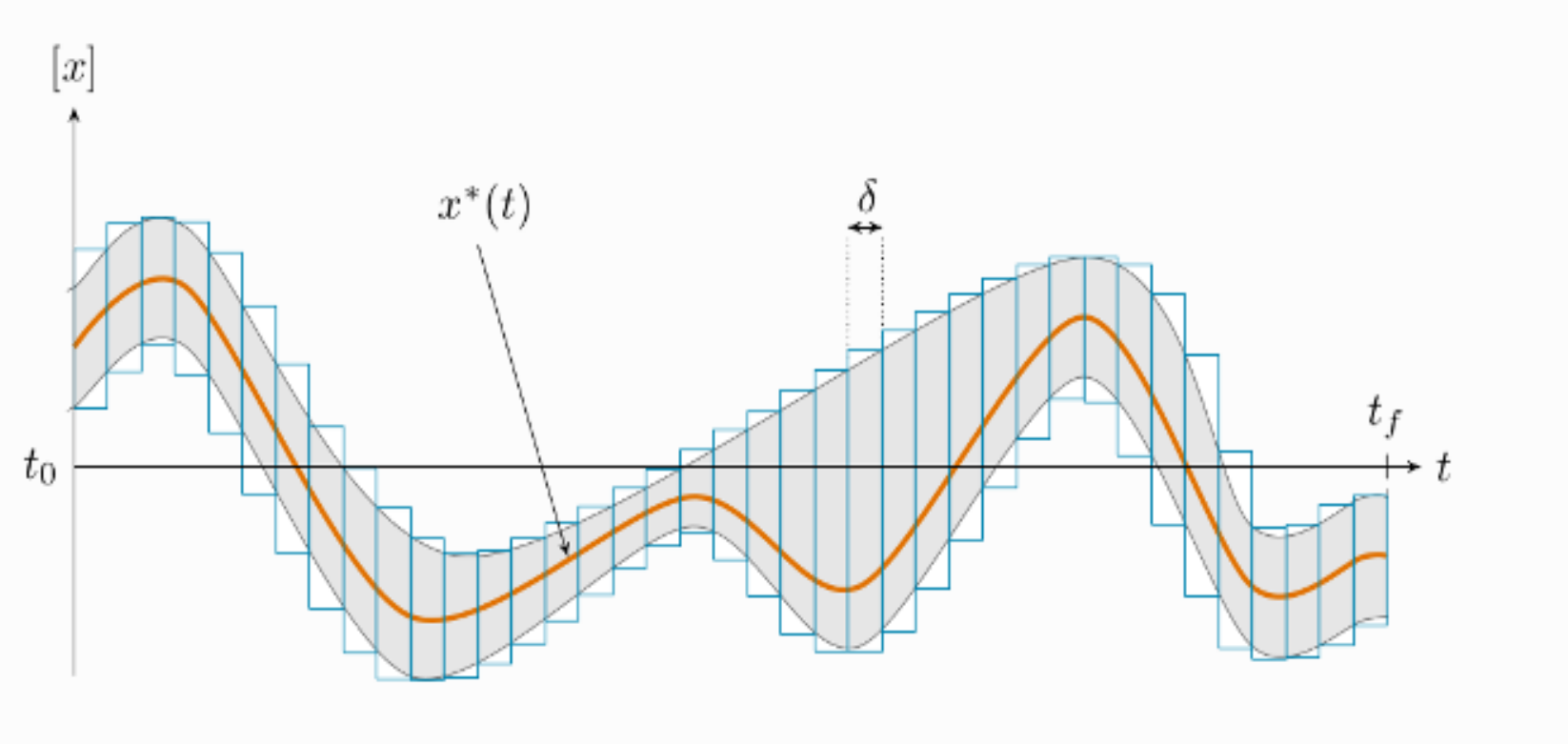
II. Travail réalisé

1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycles stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

Librairie Tubex



Tube



Stabilité des cycles par Tubex

- Définition des ensembles de départ
- Application d'un cycle L aux petites boîtes représentant l'ensemble de départ
- Vérification que le cycle passe bien par nos isobathes.
- Affichage des iso-bathes et des différents tubes représentant l'évolution de nos boîtes initiales

II. Travail réalisé

1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycle stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

Machine à états finis

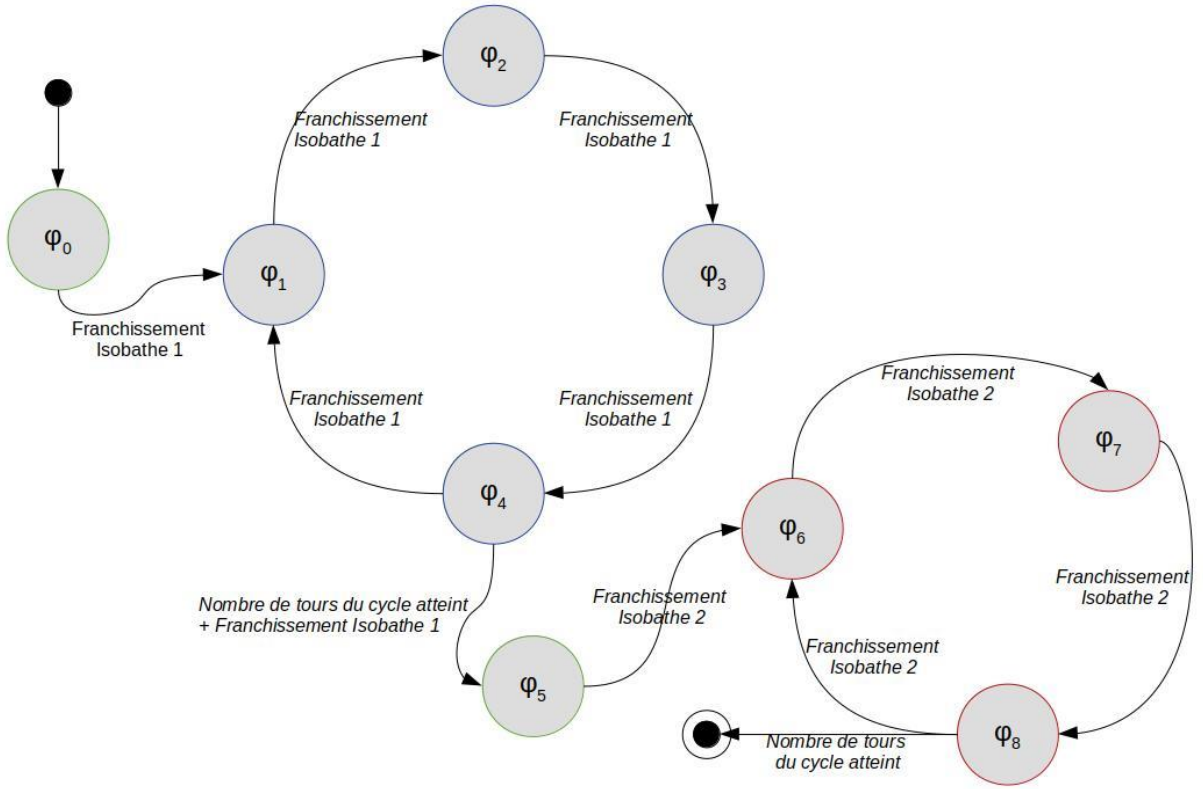
Implémentation d'une machine à états finis dans le contrôleur :

- **Paramètres :** liste des caps des différents cycles
- **Entrée :** profondeur
- **Sortie** : cap

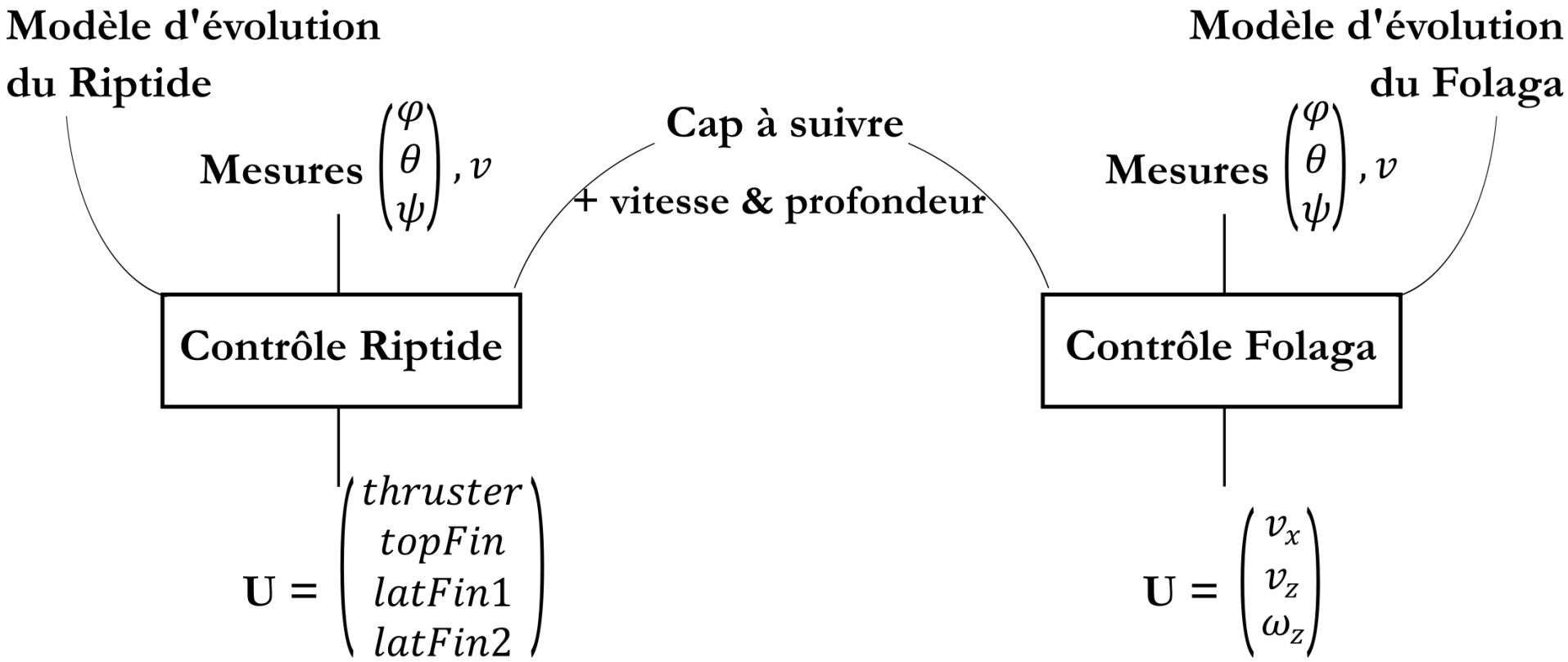
Définition de la **fsm** dans le code par des **transitions** (états de départ, états d'arrivée, fonctions booléennes de condition, fonctions appelées)

Travaux sur l'automatisation de la génération de l'automate.

Diagramme d'états-transitions de deux cycles.



Contrôle en vitesse et attitude

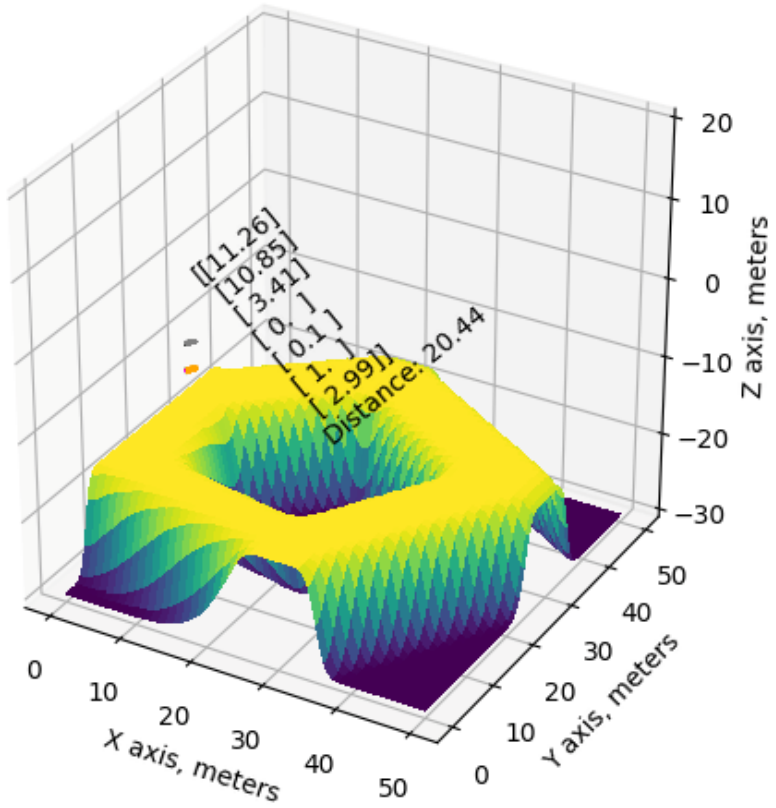


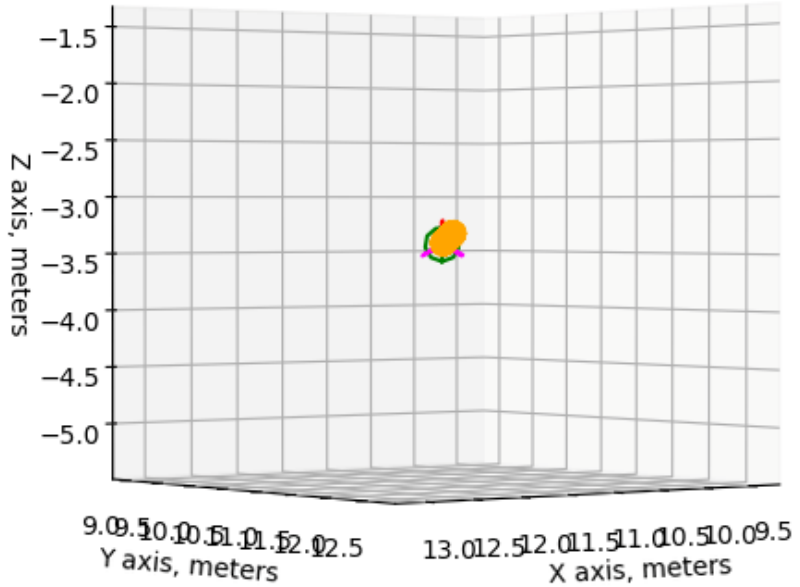
Validation du contrôle et des estimations

$$\dot{v} = p_1 . u_0^2 - p_2 . v^2$$
$$\Rightarrow u_0 = \sqrt{\frac{\widehat{p_2}}{\widehat{p_1}}} . v_{target}$$

$$\begin{pmatrix} \dot{\varphi} \\ \theta \\ \psi \end{pmatrix} = v^2 . E'(\varphi, \theta, \psi) . B \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

$$\Rightarrow U = \frac{1}{\hat{v}^2} . B^{-1} . \hat{E'}^{-1} . K . \begin{pmatrix} \varphi_{target} - \hat{\varphi} \\ \theta_{target} - \hat{\theta} \\ \psi_{target} - \hat{\psi} \end{pmatrix}$$





II. Travail réalisé

1. Utilisation des AUV
    1. Riptide
    2. Folaga
2. Modélisation des AUV
    1. Modèles des AUV
    2. Identification des paramètres
3. Gestion des cycles stables
    1. Trouver des cycles stables
    2. Intelligence de parcours entre cycle stables
    3. Stabilité des cycles par Tubex
4. Simulation
    1. Simulation de la commande
    2. Simulateur 3D

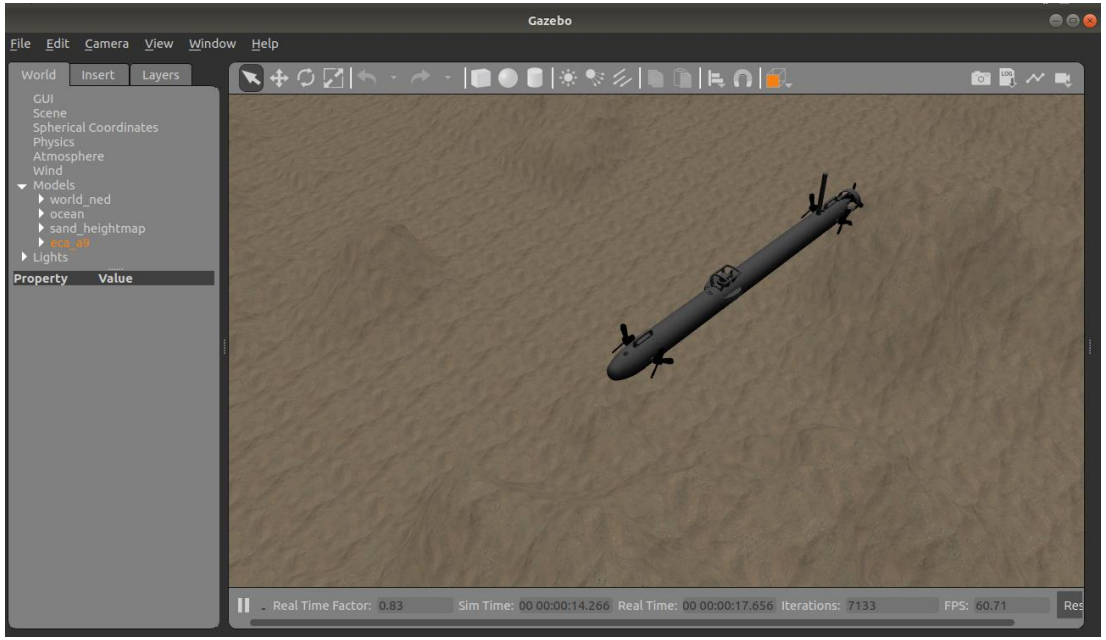
Gazebo

- Simulateur 3D utilisant un logiciel tier
- Fonctionne en parallèle de ROS
- Simulation réaliste de la physique

- Développement de ses propres robots
    - Choix de l'environnent (world)
    - Format URDF
    - Implémentation de capteurs
    - Simulation de l'environnement
- Utilisé dans plusieurs compétition

Création de la simulation

1. Choisir son environnent
2. Créer son Robot
3. Implémenter les capteurs/actionneurs
    1. Choix du Plugins
    2. Définir la particularité du capteur/actionneurs
4. Implémenter la physique

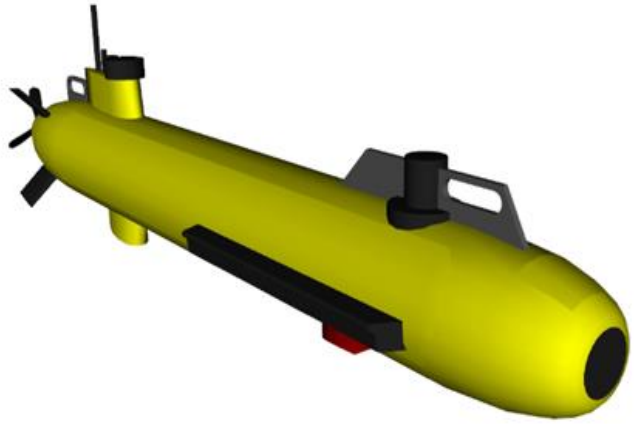


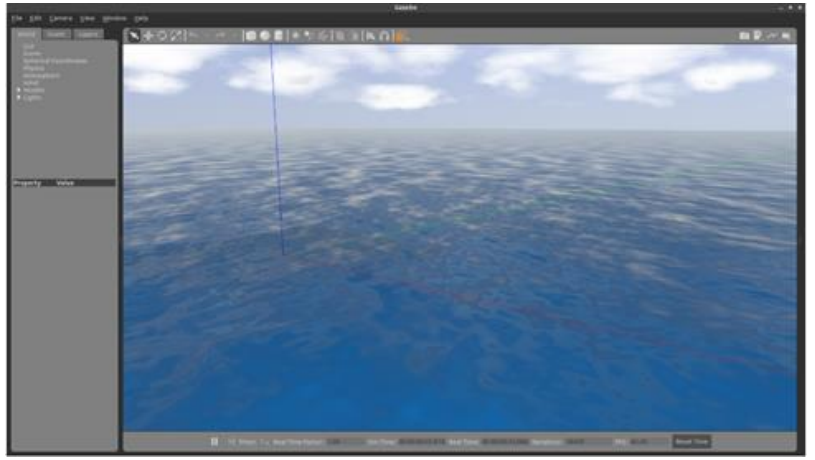
UUV Simulator

Gazebo/ROS : Simulation propulseur et ailette, modèle hydrodynamique

Gazebo worlds : Océan, lac etc ...

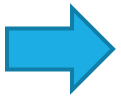
Modèles existants : AUV (eca_a9), ROV.





Modèle hydrodynamique

- Equations de Fossen
- Matrice de masse ajoutée
- Matrice Linear Damping pour simuler les frottements
- Poussée d'Archimède



Comportement du Folaga cohérent

III. Expérimentation

1. Préparation de la mission
   a. Lieu et Date
   b. Préparation logicielle
   c. Cycles stables

2. Exécution de la mission
   a. Manipulations préliminaires
   b. Réalisation des cycles stables

3. Résultats

III. Expérimentation

1. Préparation de la mission

a. Lieu et date

- **Penfeld** : pas de MNT 
- **Ty Colo**: MNT 

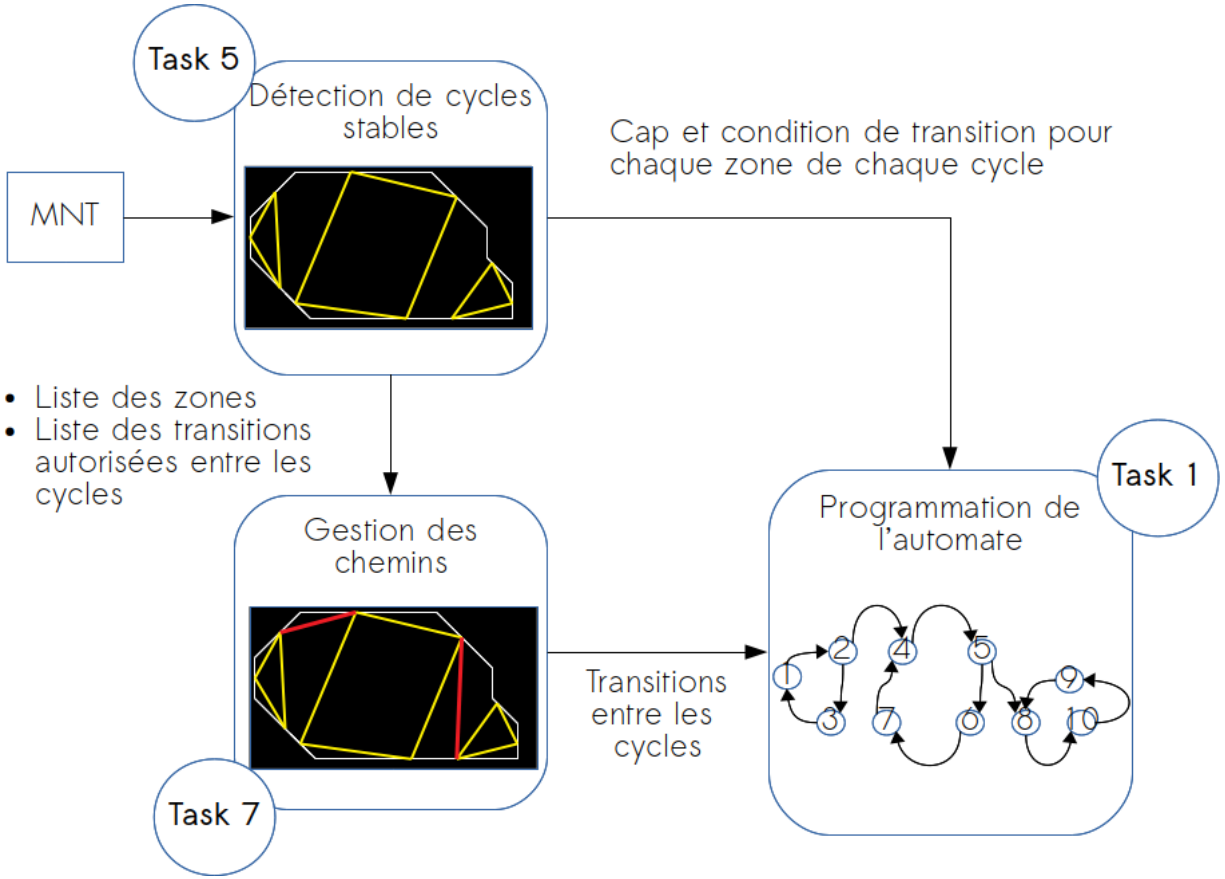
Date: Lundi 24 Février 2020

III. Expérimentation

1. Préparation de la mission

b. Préparation logicielle

- Diffusion de l'information doublée
- Programmation de l'automate nécessite du temps

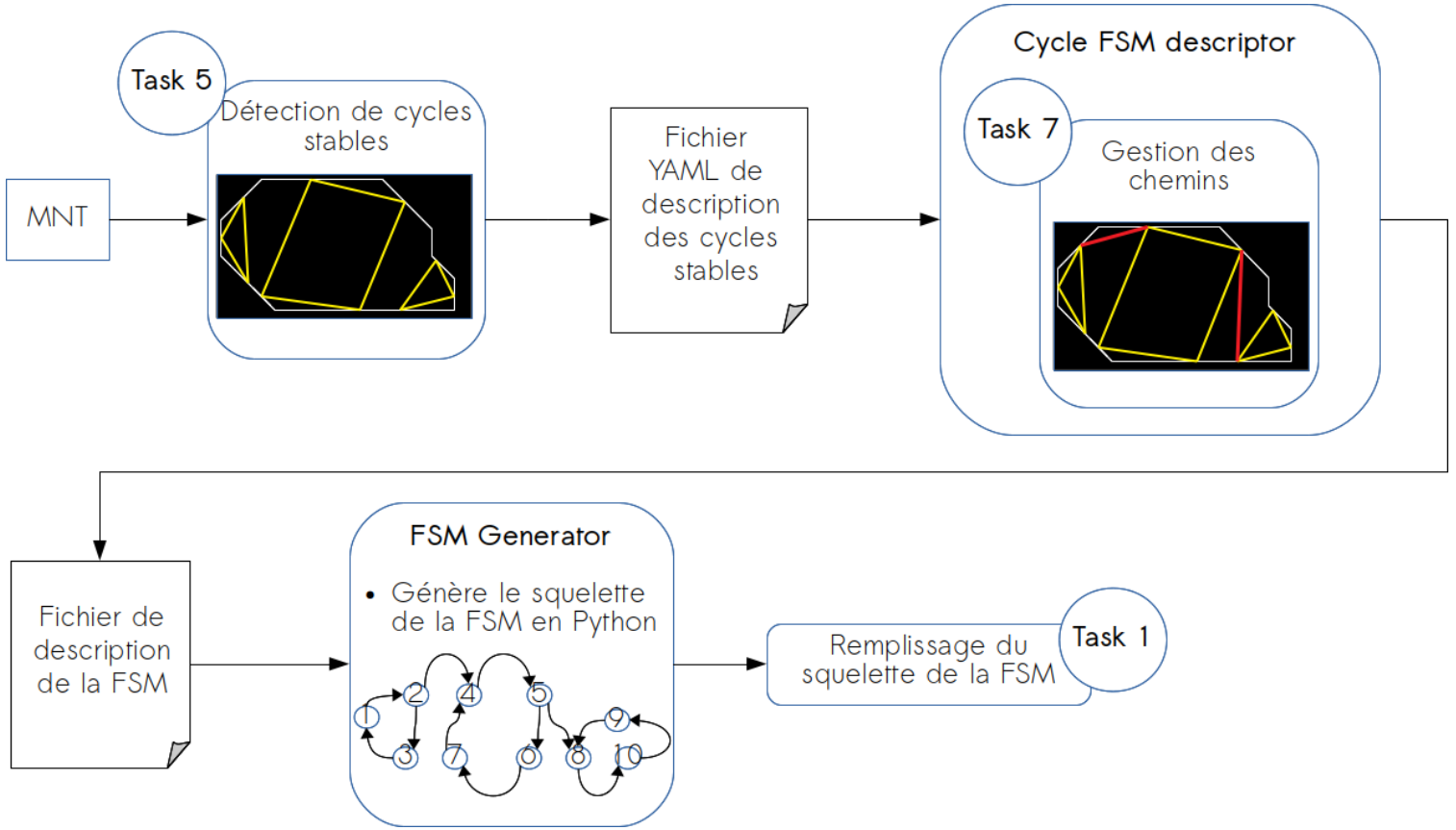


III. Expérimentation

1. Préparation de la mission

b. Préparation logicielle

- Normalisation des informations diffusées
- Automatisation partielle de la programmation de l'automate

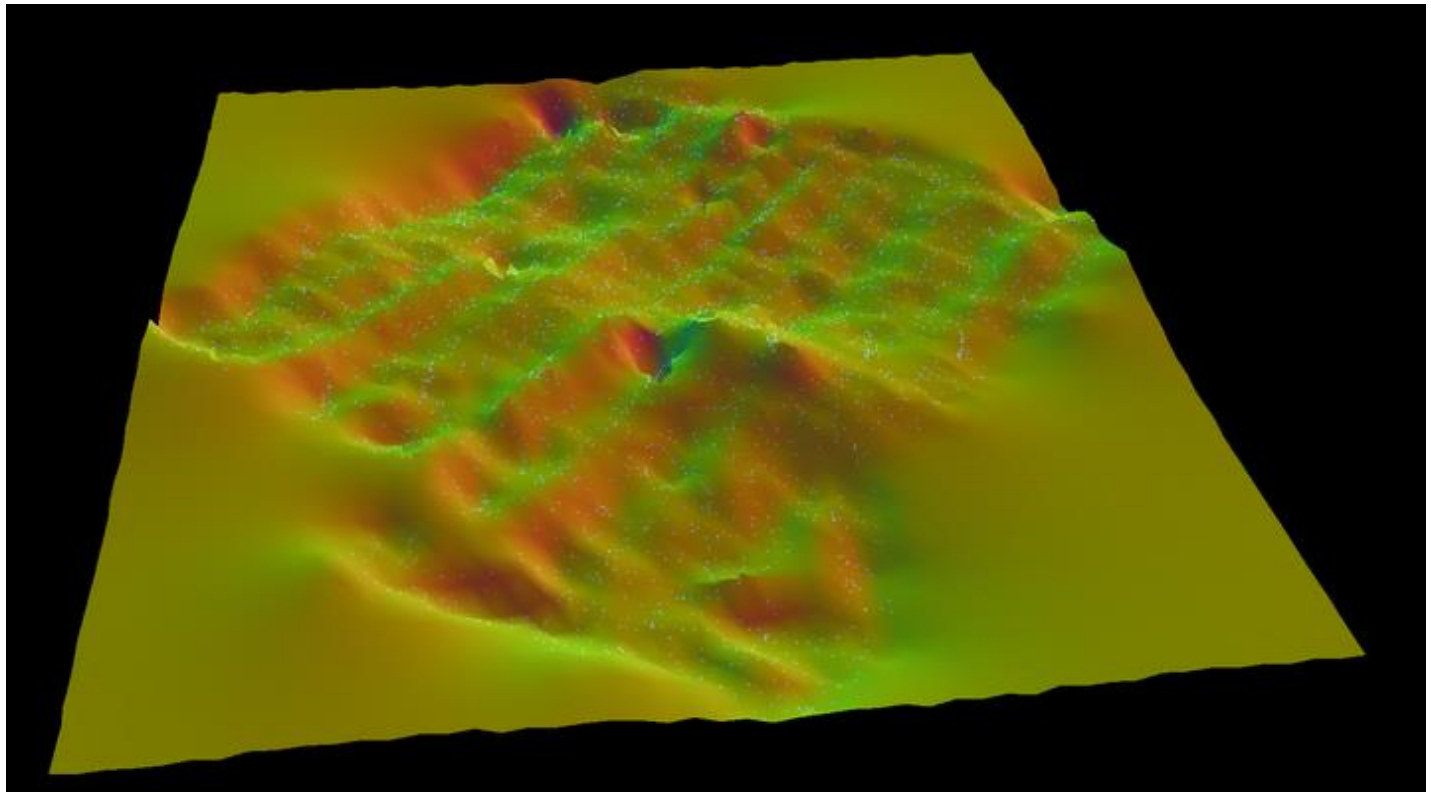


III. Expérimentation

1. Préparation de la mission

c. Cycles stables

- MNT non exploitable



III. Expérimentation

1. Préparation de la mission

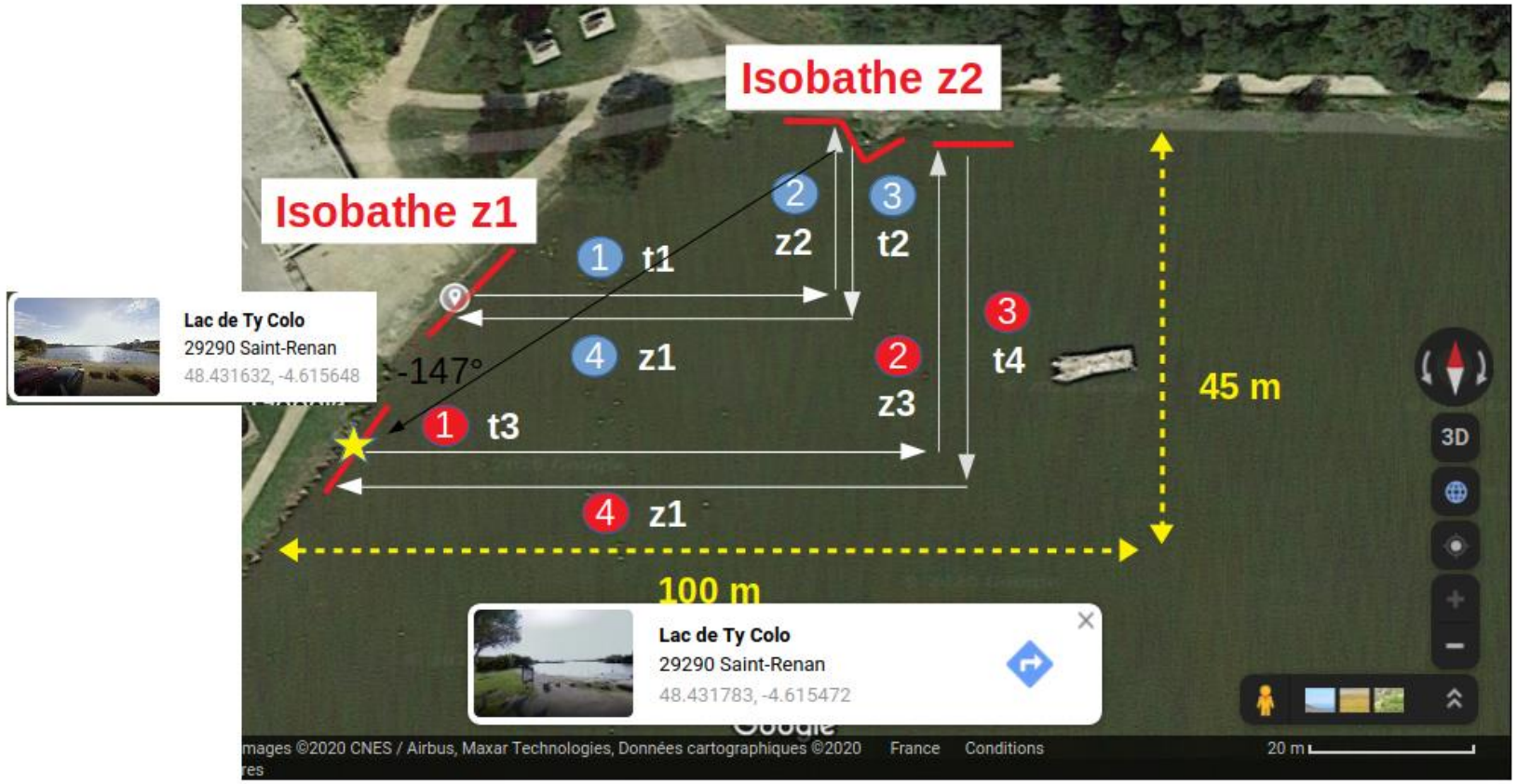
c. Cycles stables

- Choix de cycles en "L"
- Stabilité prouvée par Tubex

III. Expérimentation

1. Préparation de la mission

c. Cycles stables



III. Expérimentation

1. Préparation de la mission

c. Machine à états

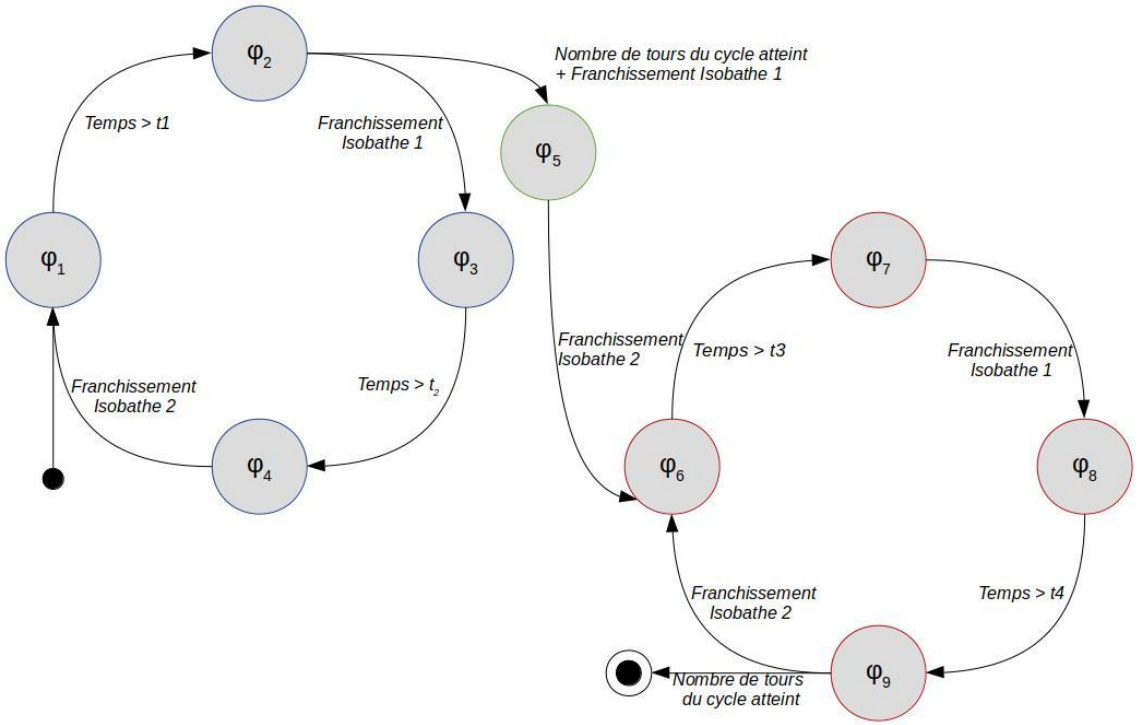
Conditions de transition sur les caps :

- Franchissement des isobathes
- Dépassement temporel

Conditions de transition sur les cycles :

- Nombre de tours de cycle

Diagramme d'états-transitions des deux cycles en "L"



III. Expérimentation

1. Préparation de la mission
   a. Lieu et Date
   b. Préparation logicielle
   c. Cycles stables

2. Exécution de la mission
   a. Manipulations préliminaires
   b. Réalisation des cycles stables

3. Résultats

III. Expérimentation

2. Exécution de la mission

a. Manipulations préliminaires

Riptide

- But : tester API constructeur
- Suivi de cap aller-retour effectué
- Pas d'immersion réalisée

Folaga :

- But : préparer les cycles en L
- Mesure des profondeurs
- Réglage de l'offset du cap

III. Expérimentation

2. Exécution de la mission

b. Réalisation des cycles stables

- 2 fois 6 petits "L"
- 2 petits "L" + 2 grands "L" + transition

III. Expérimentation

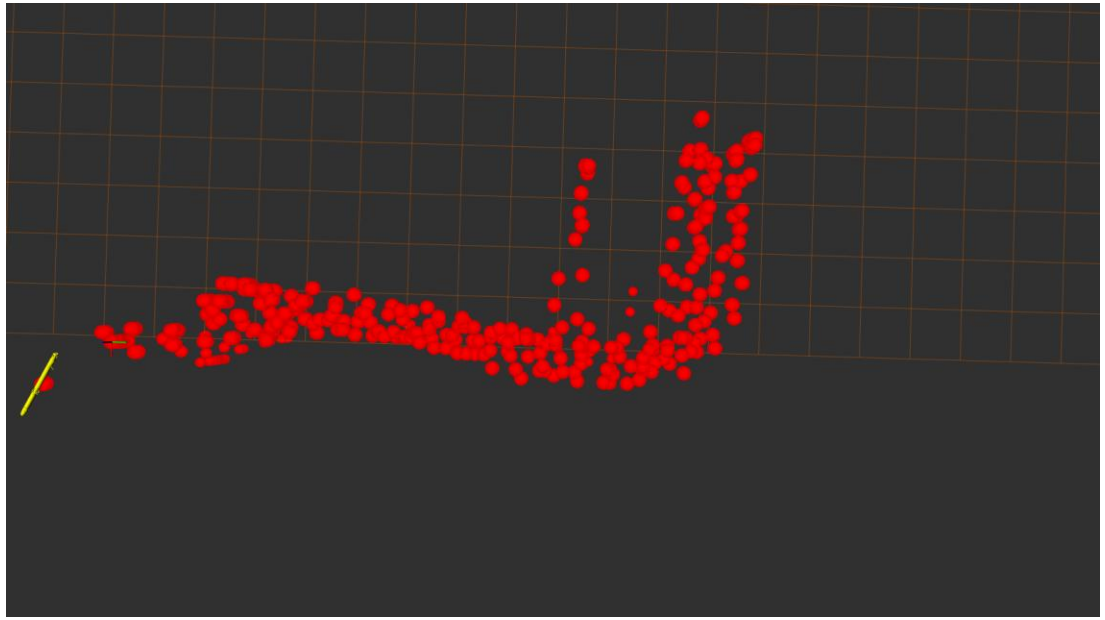
1. Préparation de la mission
   a. Lieu et Date
   b. Préparation logicielle
   c. Cycles stables

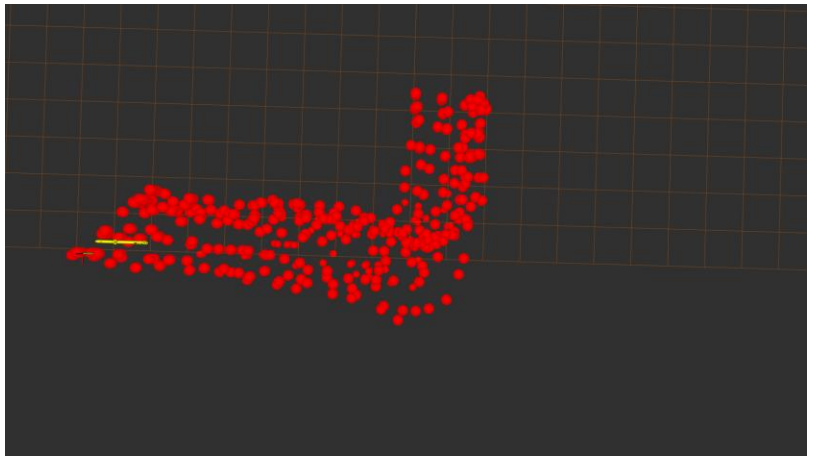
2. Exécution de la mission
   a. Manipulations préliminaires
   b. Réalisation des cycles stables

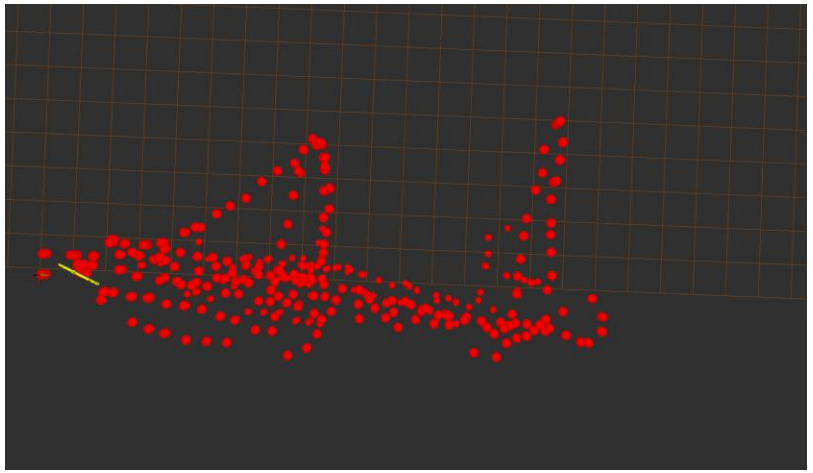
3. Résultats

III. Expérimentation

3. Résultats







III. Expérimentation

3. Résultats

- Utilisation des données pour estimation des paramètres

- Utilisation des données pour tester le simulateur 3D

Conclusion

Conclusion

Travail de groupe réussi:

- Cohérence entre les différentes tâches
- Bonne organisation et gestion des imprévus

Cycles stables : preuve de concept validée

Mais non expérimenté à partir d'un MNT

Riptide : prise en main partielle

Evolutions et améliorations possibles dans les travaux effectués