

STAGE ASSISTANT INGÉNIEUR

Exploration d'une zone karstique à l'aide d'un robot sous-marin



Stagiaire	Paul-Antoine LE TOLGUENEC (FISE 2021 - ROB)
Maitre de stage	Gilles TROMBETTONI
Doctorant	Verlein RADWAN
Professeur encadrant	Luc JAULIN

October 14, 2020



Contents

1	Remerciements	3
2	Introduction	4
3	Mise en contexte	5
3.1	Présentation de l'entreprise	5
3.2	Les enjeux du projet	5
3.3	Gestion du projet	6
4	Présentation du robot	7
4.1	Milieu d'évolution du robot	7
4.2	Le robot	9
4.2.1	L'architecture du robot	9
4.2.2	Les capteurs	10
5	Simulateur	11
5.1	Spécifications du simulateur	11
5.2	Architecture logicielle	13
5.3	Implémentation du simulateur	13
5.3.1	UUV simulator	13
5.3.2	Conception d'un monde	15
5.3.3	Simulation opérationnelle	16
5.3.4	Plateforme Open Source	17
6	Localisation du robot au sein de la karst	17
6.1	Etat de l'art des méthodes de SLAM indoor	18
6.2	L'implémentation d'un SLAM en Intervalle	20
7	Scan Matching	22
7.1	Modélisation basique	22
7.2	Matching	24
8	Estimation de l'évolution	26
8.1	Problème de minimisation	27
8.2	Contraction	27
8.3	Système de coordonnées polaires	30
8.4	Évolution envisagée	32

9 Conclusion	32
9.1 Simulateur	32
9.2 ICP en analyse par intervalle	33

List of Figures

1 Cycle agile	6
2 Représentation approximative du réseau karstique [1]	7
3 Exemple d'une zone karstique	8
4 Robot téléopéré H1000 de la société ECA Robotics	10
5 Mesure sonar acoustique wide-beam	11
6 Diagramme pieuvre du simulateur	12
7 Graphe des noeuds de la simulation affiché avec RQT	14
8 Exemple de grotte réalisée avec blender	15
9 La grotte finale intégrée à la simulation	16
10 Affichage de la trajectoire avec RVIZ.	17
11 Exemple méthode ICP	19
12 Graph-Based SLAM	20
13 Schéma appareillage point vue depuis deux repères différents.	21
14 Modèle simplifié	23
15 Mesures prises par le robot aux instants T1 et T2	24
16 Toutes les mesures ramenées dans le repère du robot en T1	25
17 En rouge le hull de l'ensemble des mesures T1 associées à une mesure T2	26
18 Toutes les mesures ramenées dans le repère du robot en T1	28
19 En noir, la partie de chaque boite verte qui est cohérente.	29
20 En noir le résultat de la contraction.	30
21 wrapping effect	31

List of Tables

1 Remerciements

Je tiens tout d'abord à remercier les équipes du Laboratoire d'informatique, de robotique et de microélectronique de Montpellier et plus particulièrement l'équipe Coconut qui m'a réservé un très bon accueil. Aussi un grand remerciement à mon tuteur de stage, le Professeur et chercheur Gilles TROMBETTONI responsable du sujet de thèse: "Intervalles, systèmes dynamiques et trajectoires garanties pour la robotique d'exploration karstique" qui m'a donné la possibilité de faire un stage aussi enrichissant sur de multiples aspects et qui a fait preuve d'une disponibilité sans faille malgré les événements actuels. Je remercie également le doctorant Verlain RADWAN qui m'a permis de mieux comprendre certains aspects théoriques de l'analyse par intervalle. Je souhaite également remercier le Professeur et chercheur Luc Jaulin, mon tuteur à l'ENSTA Bretagne, qui m'a offert l'opportunité d'effectuer ce stage.

2 Introduction

Les robots d'exploration dans un environnement hostile restreint sont actuellement confrontés à d'énormes défis, non seulement dues à la diversité de l'environnement cible, mais aussi à la nécessité de garantir l'exactitude des données collectées. Aleyin (Robotique d'Exploration Karstique) est un projet porté par l'Université de Montpellier en lien avec le sujet proposé. Il vise à développer un système robotique en mesure d'acquérir les données nécessaires à la caractérisation de la dynamique de réseaux karstiques, lieu de la ressource en eau. Le projet RS Région Occitane, LEZ-2020, en est issu. La cartographie de ces réseaux présente un intérêt majeur pour la gestion active de la ressource en eau, la prévision et la prévention des risques hydrologiques [2]. Mais ces données ne sont pleinement exploitables que si elles s'accompagnent de garanties. C'est pourquoi une équipe de chercheurs du LIRMM tente de mettre au point une nouvelle technique de SLAM volumétrique utilisant l'analyse par intervalle. Cette modélisation mathématique devrait permettre ces garanties, de par son approche ensembliste. Durant ce stage j'ai eu pour mission de travailler en lien avec Gilles TROMBETTONI et Verlein RADWAN. Monsieur TROMBETTONI et Monsieur RADWAN n'étant pas accoutumés au monde de la robotique, ma première mission a été de réaliser un simulateur leur permettant de tester leurs algorithmes de SLAM dans des conditions réalistes. Durant la deuxième partie de mon stage, nous avons tenté de mettre au point une nouvelle méthode ICP (Iterative closest point) avec une approche intervalle.

Repo GitHub

https://github.com/Paul-antoineLeTolguenec/karst_simulation
<https://github.com/Paul-antoineLeTolguenec/scan-matching-using-interval-analysis>

3 Mise en contexte

3.1 Présentation de l'entreprise

Le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier - LIRMM - est une unité mixte de recherche, dépendant conjointement de l'Université Montpellier et du Centre National de la Recherche Scientifique. Ses activités de recherche positionnent pleinement le LIRMM au coeur des sciences et technologies de l'information, de la communication et des systèmes. Ainsi, de l'information aux systèmes, de la technologie à l'humain et aux usages, les activités de recherche du LIRMM concernent : la conception et la vérification de systèmes intégrés, mobiles, communicants, la modélisation de systèmes complexes à base d'agents, les études en algorithmique, bioinformatique, interactions homme-machine, robotique, etc. Les travaux sont menés dans trois départements scientifiques de recherche, eux-mêmes organisés en équipes :

- le département Informatique
- le département Microélectronique
- le département Robotique

J'ai effectué mon stage au sein du département Informatique intégré à l'équipe de programmation par contraintes.

3.2 Les enjeux du projet

Montpellier est l'une des villes de France dont la croissance démographique est la plus élevée. Avec sa population municipale de plus de 285 000 habitants, la ville ne cesse d'accroître ses besoins en eau. En 2019, la région du Languedoc Roussillon devient la région la plus aride de France. L'acheminement d'eau potable dans la région devient une nécessité. En 2012, une plongée sous-marine près des sources du Lez permet de découvrir une importante galerie karstique. Cette galerie n'a pas pu être cartographiée. Envoyé des plongeurs dans un environnement aussi hostile devient trop dangereux. En effet, dans cet environnement confiné la moindre gerbe de terre soulevée pourrait aveugler totalement les plongeurs. C'est pourquoi la robotique est

favorisée. Cette solution devrait permettre d'obtenir la cartographie de la karst et bien d'autres donnée sur l'eau qu'elle contient.

3.3 Gestion du projet

L'objectif de mon stage est d'apporter une première vision à la thèse que commence Verlain RADWAN en Octobre 2020. Le projet n'ayant pas encore commencé, l'organisation reste encore à établir. C'est pourquoi nous avons utilisé les cycles de la méthode agile pour avancer efficacement dans le projet.

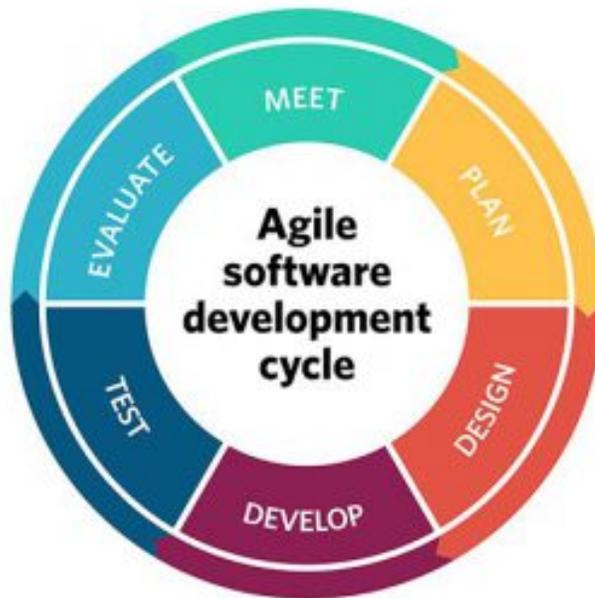


Figure 1: Cycle agile

Nos réunions étaient rythmées par les cycles de développement. J'ai eu la chance de pouvoir être intégré au LIRMM donc la majeure partie des réunions se faisaient en présentiel. A chaque fin de cycle nous faisons une réunion pour conclure sur les résultats de la fonctionnalité développée et décider quelle serait la suite.

4 Présentation du robot

4.1 Milieu d'évolution du robot

L'objectif est d'envoyer un robot dans une karst qui est en fait une grotte sous-marine. Une fois le robot dans la karst, il doit être capable de se localiser et de cartographier celle-ci. Dans un premier temps il convient de visualiser l'environnement. En effet, le fait d'avoir une idée de la géométrie de l'environnement permet éventuellement de poser certaines hypothèses sur celle-ci. Cela permet aussi d'avoir une idée des pistes à développer ou non. Ci-dessous on peut voir une vue très schématique de la zone karstique qui sera à l'étude.



Figure 2: Représentation approximative du réseau karstique [1]

En robotique, certains points géographiques par lesquels le robot passe plusieurs fois permettent de recaller la position du robot. Il est donc important de savoir si une karst peut boucler et donc si le robot peut passer deux fois par un même point géographique. Ici le manque de connaissance

de la karst impose de se placer dans le cas le plus défavorable. On fait donc l'hypothèse que le robot est en total exploration du début de la mission jusqu'au moment où il fait demi-tour. Il faut donc réaliser une méthode de SLAM extrêmement robuste. Le fait qu'aucun point remarquable ne peut être identifié implique de réaliser un SLAM volumétrique. Le SLAM volumétrique est très utilisé et développé en Indoor. En indoor on peut généralement poser certaines hypothèses telles que la continuité des murs par exemple. Dans notre cas de figure, où la géométrie de la karst est très chaotique il est difficile de poser la moindre hypothèse.

La photo figure 3 est une photo prise par des plongeurs dans une zone karstique. Comme on peut le voir, le milieu est très confiné. Le robot doit donc être pourvu d'une mobilité importante. Le projet n'étant qu'à son commencement et le robot étant en cours de développement, tous ces critères m'ont permis de visualiser l'architecture que le robot doit prendre du point de vue de la simulation.



Figure 3: Exemple d'une zone karstique

4.2 Le robot

Le projet en est à ses débuts. Le robot est donc en cours de développement. Mon objectif n'était pas de trouver toutes les solutions techniques nécessaires pour la mission du robot. Mon objectif est de faire une simulation au plus proche de la réalité avec les informations déjà établies. La vocation de cette simulation n'étant pas de faire de la commande mais plutôt d'exploiter les données générées par les capteurs du robot en simulation. J'ai pu faire une réunion avec le chercheur du LIRMM Yohan BREUX qui m'a permis de faire un premier listing des capteurs envisagés.

4.2.1 L'architecture du robot

L'architecture du robot et les actionneurs restent aussi à mettre en place. J'ai donc réalisé un état de l'art des solutions techniques utilisés dans le monde de l'industrie pour effectuer des missions similaires. Le fait de trouver un robot correspondant aux critères de la karst m'a permis d'avoir une certaine ligne directrice. Comme nous l'avons vu précédemment la géométrie de la Karst impose au robot d'avoir une grande mobilité. Il est évident que le robot doit être muni de plusieurs propulseurs pour lui permettre cette mobilité. Une architecture pertinente est donc celle du robot téléopéré H1000 de la société ECA robotics (qui lui évoluera en autonomie).



Figure 4: Robot téléopéré H1000 de la société ECA Robotics

4.2.2 Les capteurs

La simulation permettra par la suite de tester les algorithmes de SLAM. La simulation repose donc entièrement sur la partie capteur. La liste des capteurs n'est pas encore totalement définie et peut évoluer en fonction des besoins techniques du projet. Une IMU sera utilisée pour permettre de donner une estimation de l'évolution du robot au sein de la karst entre deux instants. Plusieurs sonars DVL (range only) seront disposés sur les différentes faces du robot. Une caméra pourrait éventuellement être utilisée pour coupler la donnée image dans la méthode ICP. Dans un premier temps l'utilisation de la caméra est notamment remise en question par le fait que le robot pourrait soulever des gerbes de terre qui rendrait l'utilisation de la caméra impossible. L'utilisation de deux sonars acoustique wide-beam permettra de réaliser un scan 3D de la grotte.

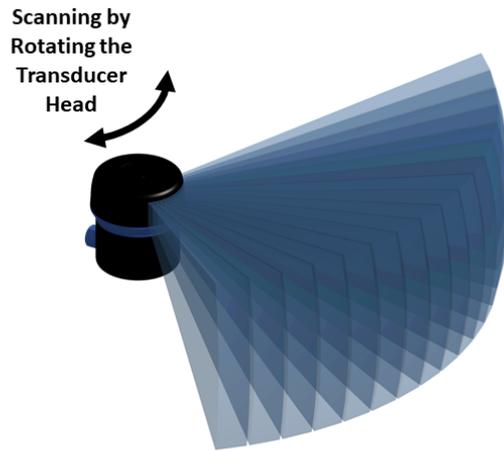


Figure 5: Mesure sonar acoustique wide-beam

5 Simulateur

5.1 Spécifications du simulateur

Pour définir au mieux les fonctionnalités du simulateur, il m'a fallu dans un premier temps déterminer quelles étaient les besoins du projet. J'ai pu établir un diagramme des spécifications fonctionnelles :

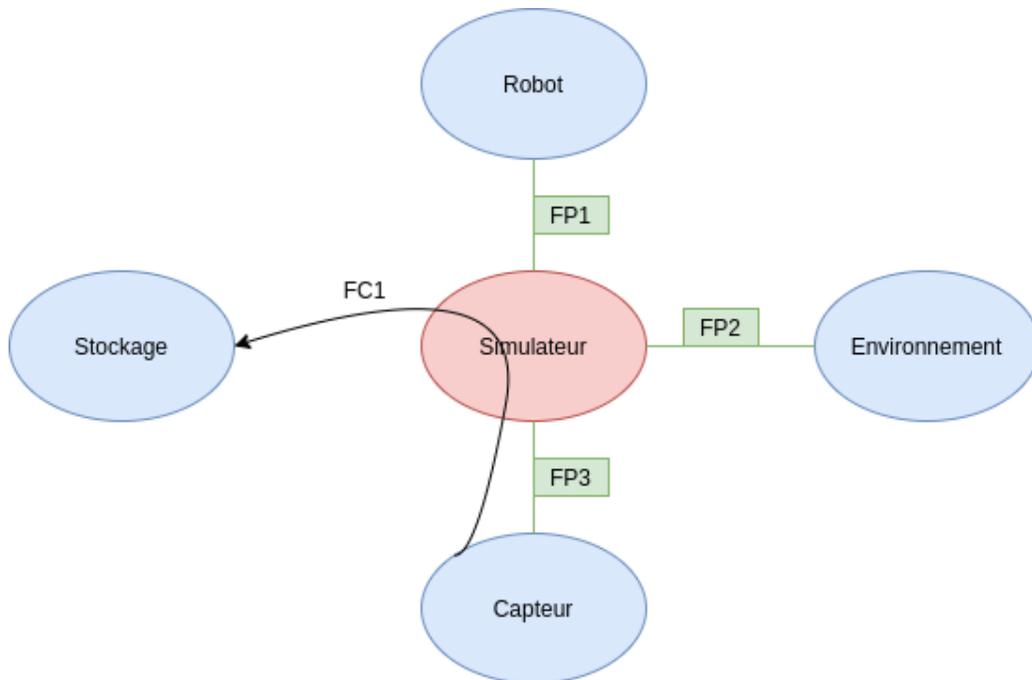


Figure 6: Diagramme pieuvre du simulateur

Ce diagramme bien qu'il n'ait pas pris cette forme durant le stage, m'a permis de suivre une certaine ligne directrice tout au long de la conception du simulateur. Ce diagramme se compose de quatre blocs principaux. La table suivante permet de détailler les fonctions principales et contraintes.

ID	Fonction
FP1	Le robot simulé doit avoir un comportement similaire à celui du robot
FP2	La simulation doit être faite dans le milieu d'évolution du robot (une Karst)
FP3	Le robot simulé doit être munie des mêmes capteurs qui équiperont le robot lors de l'exploration
FC1	La simulation doit pouvoir être re-joué sans l'interface graphique

5.2 Architecture logicielle

L'architecture logicielle a été très rapidement établie. Dans un premier temps il a fallu choisir le logiciel permettant de générer la simulation 3D. Il existe bon nombre de logiciels permettant de faire de la simulation 3D. Mais l'un des plus utilisés aujourd'hui en robotique est Gazebo. Gazebo Project, est un simulateur 3D, cinématique, dynamique et multi-robot permettant de simuler des robots articulés dans des environnements complexes, intérieurs ou extérieurs, réalistes et en trois dimensions. Le fait qu'il s'agisse d'un programme open source implique qu'une grande communauté de roboticien partage les travaux qu'ils ont réalisés sur Gazebo. Il n'est donc pas toujours nécessaire de partir de zéro. Ensuite, il fallait utiliser un middleware pour interfacier la simulation avec nos codes. En effet, Gazebo permet de faire de la simulation tout comme un "video game engine". Nous avons utilisé le middleware ROS. Robot Operating System (ROS), est un ensemble d'outils informatiques open source permettant de développer des logiciels pour la robotique. L'avantage est qu'il est possible d'interfacier Gazebo avec ce middleware. Le LIRMM développe aussi son propre middleware. Le robot final tournera avec le middleware développé par le LIRMM. Mais ce middleware n'étant pas encore opérationnel nous avons décidé d'utiliser ROS.

5.3 Implémentation du simulateur

Le simulateur Gazebo offre une structure riche qui permet de travailler en agile. La création des éléments de la simulation se fait en éditant des fichiers au format xacro. Il est donc intéressant de trouver des simulations open source et partir de celles-ci. Sur le site ROS WIKI on peut trouver un listing des simulations que le middleware approuve. L'une des simulations les plus utilisées pour le milieu marin est le package UUV Simulator.

5.3.1 UUV simulator

Le simulateur UUV simulateur [3] est un ensemble de modules comprenant des plug-ins et des applications ROS qui permettent de simuler des véhicules sous-marins dans Gazebo. L'ensemble de ces packages est disponible sur

le compte GitHub du projet. Le projet est extrêmement riche de fonctionnalités avec la possibilité d'utiliser différents robots pour une simulation et des contrôleurs différents pour chaque robot. En revanche, le point très négatif du package est que la documentation est très peu détaillée. Dans un premier temps j'ai donc dû prendre en main le simulateur avec des simulations basiques.

Le package fonctionne avec ROS. Connaissant bien le middleware, j'ai pu mieux comprendre l'architecture du projet grâce au graphe de noeuds que l'on peut visualiser avec RQT. Ce graphe des noeuds ROS est essentiel pour comprendre la simulation. Il m'a permis par la suite de déterminer les contrôleurs à utiliser et les topics sur lesquels publier. En effet, plutôt que de créer un nœud de commande j'ai utilisé des contrôleurs déjà réalisés sur ROS.

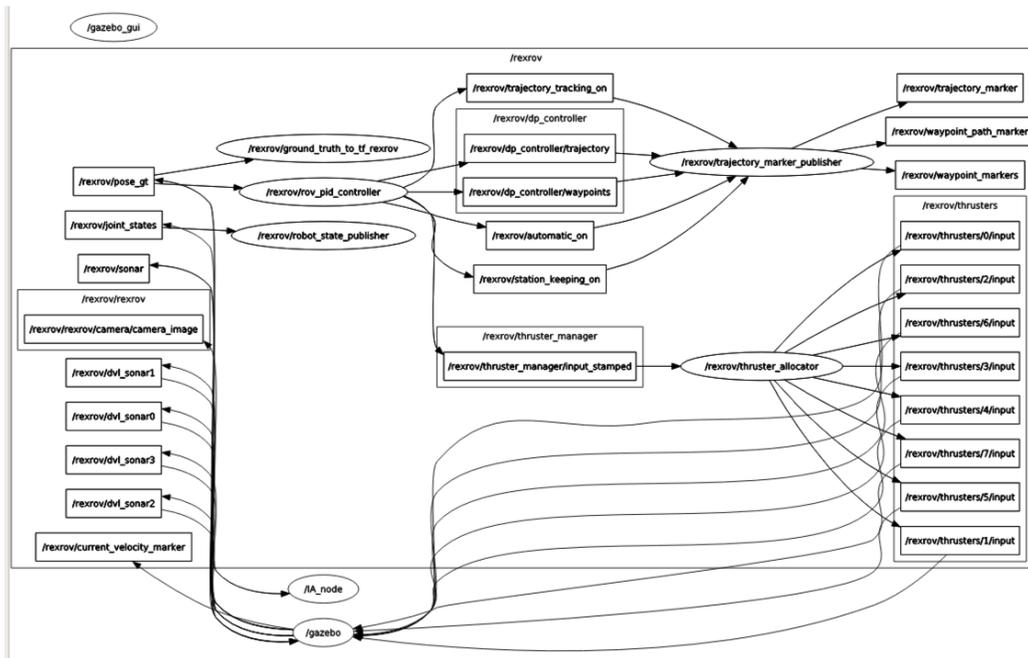


Figure 7: Graphe des noeuds de la simulation affiché avec RQT

5.3.2 Conception d'un monde

La première étape a été la conception d'un monde. J'ai créer le "seabed" avec un fichier script python. Cela permet d'obtenir le sol du monde. Il m'a fallu créer un monde ressemblant au mieux au milieu d'évolution du robot. Pour créer ce qui ressemble à une karst, j'ai dû utiliser un logiciel d'édition 3D Blender. Blender est un logiciel libre de modélisation, d'animation par ordinateur et de rendu en 3D. Ce logiciel permet de faire des modélisations très réalistes.

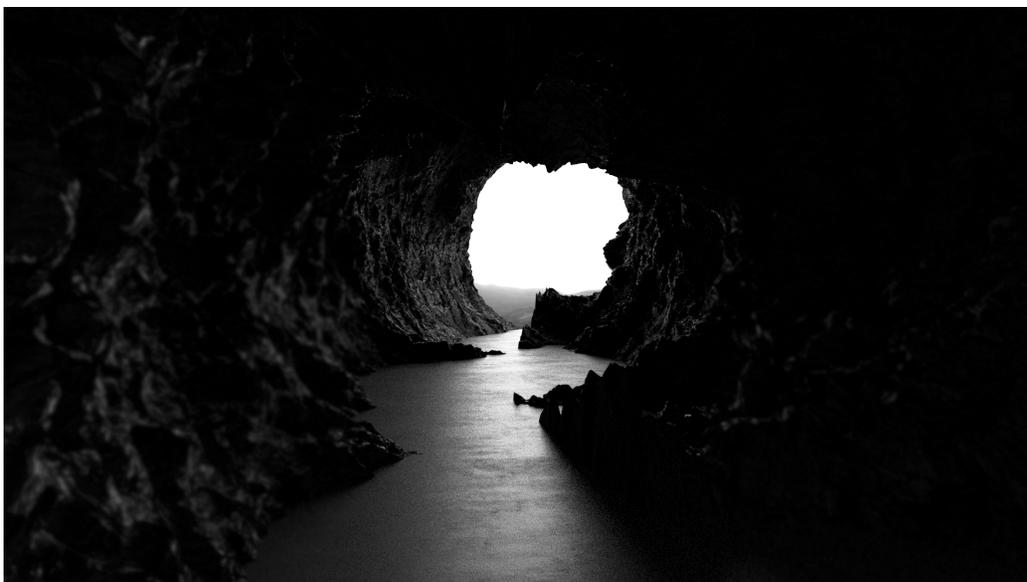


Figure 8: Exemple de grotte réalisée avec blender

Un détail à prendre en compte est la taille du fichier blender exporté dans le monde. En effet, si les détails de la grotte sont trop précis, la simulation peut être très énergivore. Il existe donc un juste milieu entre la qualité des graphismes et une simulation économe en énergie. [4].

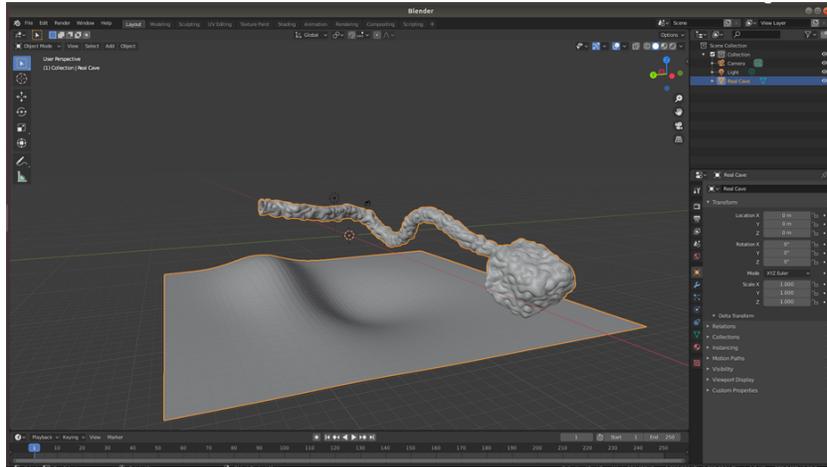


Figure 9: La grotte finale intégrée à la simulation

5.3.3 Simulation opérationnelle

L'objectif est de rendre la simulation facile d'utilisation notamment pour des personnes qui ne connaissent que très peu ROS et Gazebo. De façon à ce que le robot suive la trajectoire de la grotte à tout moment, j'ai utilisé un contrôleur qui permet de faire suivre un point GPS au robot. Pour déterminer les points GPS de la grotte j'ai utilisé un script python lancé depuis le terminal Blender. Le script permet de déterminer les barycentres des formes basiques (cylindres) qui composent la Karst. On obtient un nuage de points. Il faut appliquer la même transformation au nuage de points qu'à la Karst (rotation, translation, rescale). De plus, J'ai créé un fichier launch édité en xml qui permet de lancer tous les packages de la simulation depuis un terminal. J'utilise RVIZ pour permettre de mieux visualiser la trajectoire établie et l'odométrie.

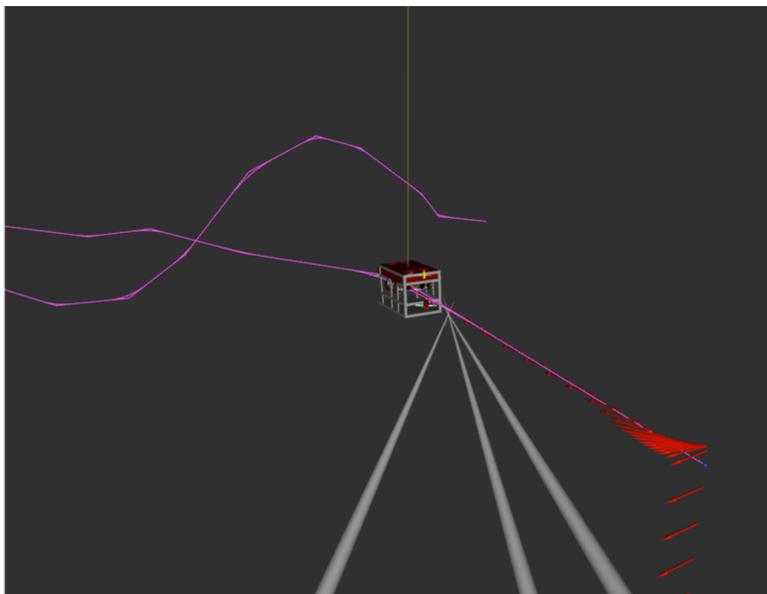


Figure 10: Affichage de la trajectoire avec RVIZ.

Enfin, de façon à rendre l'utilisation de ROS plus simple. J'ai réalisé un noeud ROS permettant de récupérer toutes les informations capteurs essentielles. Ce noeud est aussi l'occasion d'intégrer les bibliothèques d'IA TUBEX et IBEX essentielles pour la suite du projet.

5.3.4 Plateforme Open Source

Une fois le simulateur totalement fonctionnel, je me suis attaché à rendre le projet facilement accessible. J'ai publié le projet sur mon compte GitHub. J'ai réalisé toute la documentation permettant aux différentes personnes de l'équipe de pouvoir installer correctement tous les packages. Certains codes du packages uuv simulator étant erronés, je les ai mis à jour sur mon GitHub.

6 Localisation du robot au sein de la karst

L'objectif principale de ce stage a été d'apporter une première vision sur le sujet de thèse de Verlain RADWAN. Mais ce sujet est extrêmement

large et complexe. La problématique globale est de réaliser un SLAM (simultaneous localization and mapping) au sein d'une Karst en utilisant l'analyse par intervalle. Mais ce problème se décompose en de nombreuses tâches qui n'ont que très peu été abordées dans le monde de la recherche. Dans un premier temps nous avons dressé un état de l'art de ce qui existe concernant les méthodes de SLAM Indoor.

6.1 Etat de l'art des méthodes de SLAM indoor

Lors de la première partie de mon stage j'ai pu m'entretenir avec Yohan BREUX chercheur du LIRMM qui développe une technique de SLAM pour le robot. Cette première approche est essentielle car cela permet de mieux comprendre quelles sont les contraintes d'analyses d'une telle technique mathématique. De nos jours les méthodes de SLAM indoor se décomposent de la sorte :

- Une première partie que l'on nomme méthode ICP (Iterative Closest Point).

L'Iterative Closest Point [5] est un algorithme utilisé dans le but de mettre en correspondance deux jeux de données, le plus souvent sous la forme de nuages de points ou maillages correspondant à deux vues partielles d'un même objet. Une vue étant constituée d'un ensemble de points, l'objectif est de minimiser itérativement la distance entre ces points. En effet, le robot est muni de capteurs. Ces capteurs de distance vont nous permettre de faire des acquisitions à chaque pas de temps. Ces données capteurs nous permettent de percevoir l'environnement du robot au moment de l'acquisition. Ainsi, si le robot bouge entre deux instants t_1 et t_2 avec une évolution q , le robot perçoit l'environnement dans une autre configuration. L'objectif de la méthode ICP est de déterminer quelle a été l'évolution q du robot entre ces deux instants. Cette méthode permet donc d'affiner la connaissance de cette évolution q qui est souvent déjà connue partiellement grâce au capteur IMU.

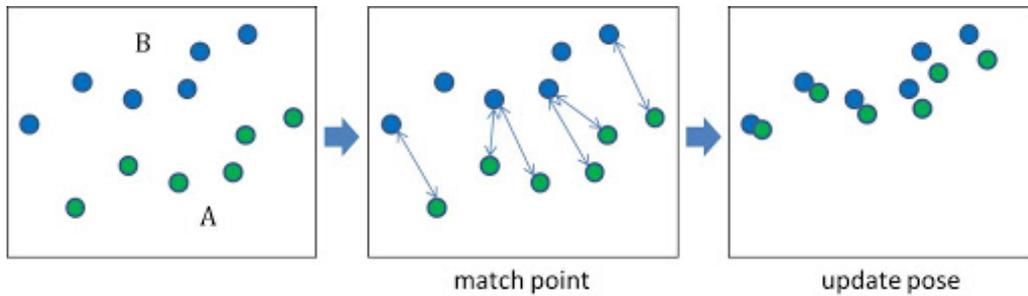


Figure 11: Exemple méthode ICP

- La deuxième partie concerne le "Graph-based SLAM" [6].
 En effet, la première partie ICP de la méthode de SLAM permet de mieux connaître l'évolution du robot entre deux instants consécutifs. Cette méthode n'est utile que pour deux acquisitions qui concerne la même partie de l'environnement. Ainsi cette méthode est très partielle. Il existe donc une méthode plus générale qui permet de se localiser dans un environnement tout en le cartographiant. Cette méthode est basée sur l'utilisation d'un graphe de poses. Ce graphe permet d'organiser la connaissance de l'environnement dans lequel évolue le robot. En effet à chaque instant le robot va associer sa perception de l'environnement à son état, et sa commande à l'état qui le précède... Tout ce qui comporte un lien direct avec son état actuel. Le graphe va établir des connexions entre deux perceptions lorsqu'elles concernent la même partie de l'environnement perçu. Enfin, par des méthodes d'optimisation de graphe on parvient à déterminer la cartographie la plus probable de l'environnement et la position la plus probable du robot.

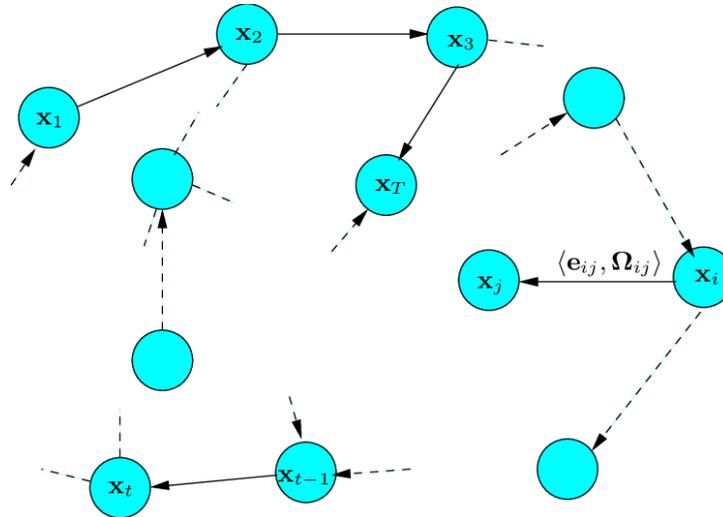


Figure 12: Graph-Based SLAM

Cette dernière méthode de graphe témoigne de la difficulté de réaliser un SLAM de qualité. En effet, la connaissance de la carte de l'environnement est extrêmement corrélée à la connaissance de la position du robot, pour connaître l'un il faut connaître l'autre correctement et inversement.

6.2 L'implémentation d'un SLAM en Intervalle

Comme nous avons pu le voir, la première phase d'un SLAM consiste à réaliser un premier traitement de l'information par la méthode ICP. Dans beaucoup d'applications, la méthode ICP permet de connaître l'évolution d'un objet dans un repère fixe à partir de deux observations de ce même objet. Dans notre cas de figure nous cherchons à connaître l'évolution du robot au sein de la karst à partir des observations de celui-ci. Ainsi, dans un premier temps notre objectif a été d'implémenter une telle méthode en utilisant l'analyse par intervalle. La méthode ICP se décompose elle-même en deux phases :

- Calcul des correspondances.
 Cette partie consiste en l'appareillage des points de deux nuages différents. En effet, la finalité est de pouvoir comparer des points entre eux. Donc il faut dans un premier temps déterminer les combinaisons de points

qui seront par la suite utilisées pour déterminer l'évolution.

- L'estimation de l'évolution.

Une fois ces combinaisons établies, une méthode des moindres carrées permet de minimiser l'erreur de l'évolution q entre les deux nuages de points et donc d'affiner la connaissance de celle-ci.

Dans les méthodes usuelles d'appareillages de points dans l'espace l'une des distances les plus utilisées est la distance de Mahalanobis. En statistique, la distance de Mahalanobis est une mesure de distance mathématique introduite par Prasanta Chandra Mahalanobis en 1936. Elle est basée sur la corrélation entre des variables par lesquelles différents modèles peuvent être identifiés et analysés. Cette mesure permet donc de déterminer une probabilité. Plus exactement, cette mesure permet de déterminer pour chaque point une ellipse contenant les points associés à ce même point. Notre objec-

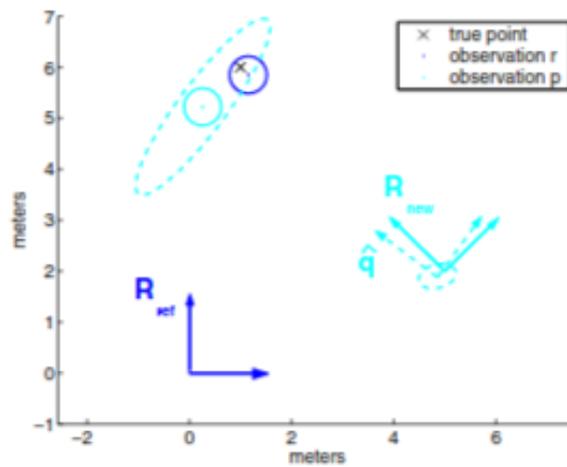


Figure 13: Schéma appareillage point vue depuis deux repères différents.

tif est donc dans un premier temps de passer d'un modèle probabiliste à un modèle ensembliste. Nous appellerons par la suite la méthode que nous avons tenté de développer le Scan Matching.

7 Scan Matching

En robotique, nous utilisons généralement une IMU (centrale inertielle) pour localiser un robot Indoor. Pour se localiser dans un tel environnement un robot ne peut pas utiliser de gps car les ondes ne passent plus à une certaine profondeur. De plus la géométrie d'une Karst étant très variable le robot doit se localiser par rapport à son environnement. L'IMU permet de renseigner avec une certaine imprécision, les vitesses et accélérations du robot. Ainsi, comme la localisation est déduite de l'intégration de données incertaines, il y a une dérive progressive dans le temps. Notre objectif est de mettre en place une méthode permettant de limiter cette dérive.

7.1 Modélisation basique

Dans le domaine de la recherche, il convient de réaliser des modélisations qui permettent de vérifier certaines hypothèses et de tester certaines solutions. Ce type de modélisation permet aussi d'apporter un support visuel parfois plus parlant que certaines équations. Etant au début du projet, nous devons avoir à disposition une modélisation très basique et en même temps réaliste. Pour réaliser cette modélisation je me suis servie de trois bibliothèques bien utilisées dans ce domaine : TUBEX, IBEX et VIBES. Pour visualiser le problème, nous avons fait une simplification de celui-ci. Nous traitons ce problème en 2D. Nous résonnons entre deux instants T1 et T2.

Le robot est muni d'une IMU. On fait l'hypothèse que nous connaissons sa position à T1 avec une certaine imprécision (représenté en rouge). Le robot évolue entre T1 et T2 avec une évolution Q estimée par l'IMU.

$$\hat{Q} = \begin{pmatrix} \hat{X}_q \\ \hat{Y}_q \\ \hat{\Theta}_q \end{pmatrix}$$

Cette estimation étant imprécise, l'erreur se propage dans la localisation du robot à l'instant T2.

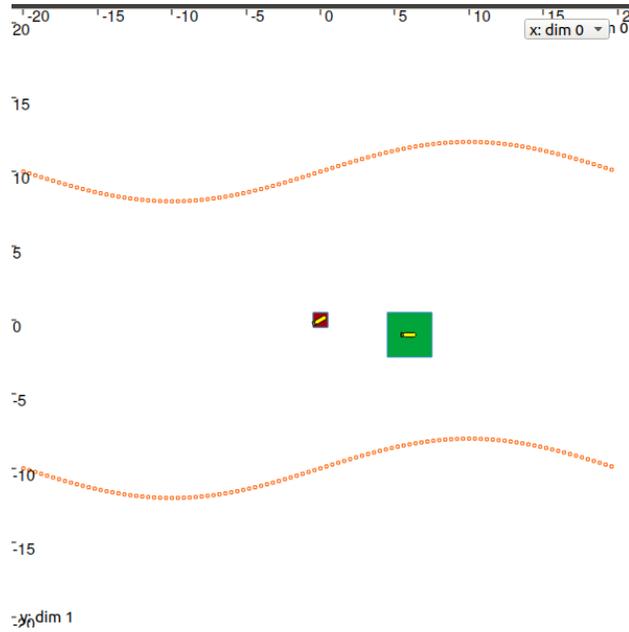


Figure 14: Modèle simplifié

Le robot est aussi muni d'un sonar qui lui permet de déterminer la distance qui le sépare de certains points qui l'entourent. Les mesures sont caractérisées par un angle et une distance comme représenté sur la figure ci-dessous. Une mesure correspond aux coordonnées polaires d'un point du mur dans le référentiel du robot. Les points identifiés à l'instant T1 sont symbolisés en rouge et ceux de l'instant T2 sont symbolisés en vert. Bien sûr le robot ne peut pas scanner deux fois le même point.

On note

$$\alpha r1i, dr1i$$

et

$$\alpha r2i, dr2i$$

l'ensemble des mesures prisent respectivement aux instants T1 et T2.

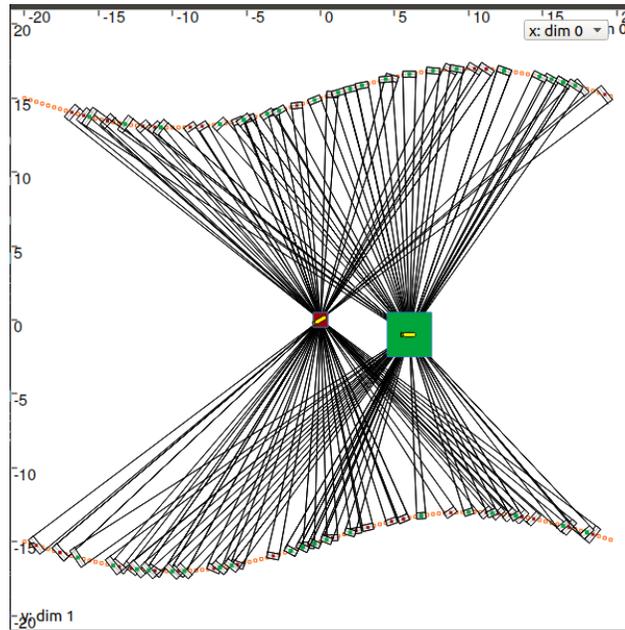


Figure 15: Mesures prises par le robot aux instants T1 et T2

7.2 Matching

La première partie du Scan Matching consiste à associer pour chaque mesure prise à l'instant T2 un ensemble de mesure prise à l'instant T1. On utilise la transformation Q du robot entre les deux instants T1 et T2 pour projeter les mesures prises à l'instant T2 dans le repère local du robot à l'instant T1. Ainsi toutes les mesures sont dans un seul et même repère, celui du robot à l'instant T1. On passe du système de coordonnées polaires au système de coordonnées cartésien. Les mesures étant initialement dans le système de coordonnées polaires dans le référentiel du robot pour T1 et T2, On utilise le contracteur $Ctcpolar$ de la bibliothèque TUBEX qui permet de réaliser cette projection. Ainsi pour toutes les mesures réalisées à T1 on applique la transformation suivante :

$$X(t) = \rho_{mes} \cos(\theta_{mes})$$

$$Y(t) = \rho_{mes} \sin(\theta_{mes})$$

Enfin pour ramener toutes les mesures prises à l'instant T2 dans le

repère du robot à l'instant T1, on applique la transformation suivante :

$$X(t) = \widehat{X}_q + \rho_{mes} \cos(\theta_{mes} + \widehat{\Theta}_q)$$

$$Y(t) = \widehat{Y}_q + \rho_{mes} \sin(\theta_{mes} + \widehat{\Theta}_q)$$

Les mesures étant toutes dans le même repère cela nous permet de les associer les unes entre les autres.

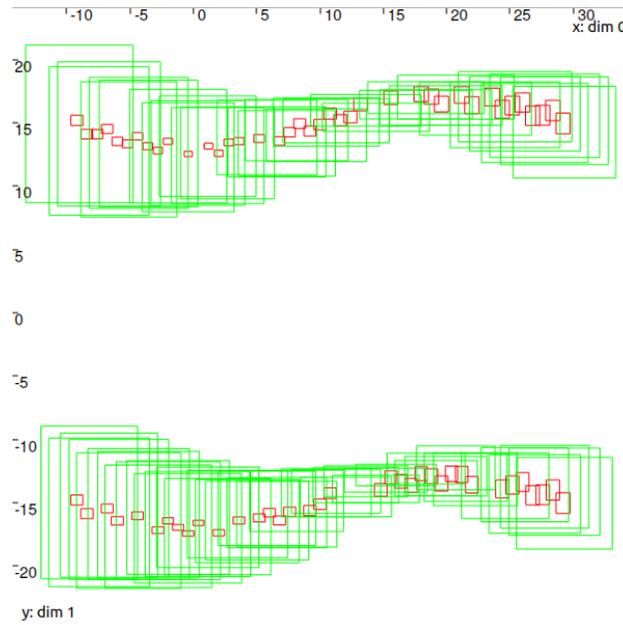


Figure 16: Toutes les mesures ramenées dans le repère du robot en T1

Pour chaque mesures vertes (T2 ramenée dans T1) on choisit d'associer les mesures rouges (mesures de T1 projetées en cartésien) dont l'intersection est non nulle.

Enfin on réalise le Hull des mesures de T1(rouge) associées à une mesure T2 (vert).

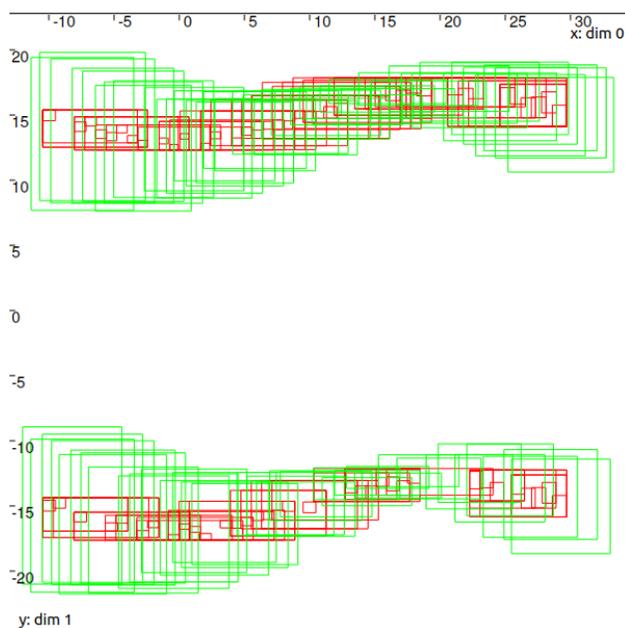


Figure 17: En rouge le hull de l'ensemble des mesures T1 associées à une mesure T2

8 Estimation de l'évolution

L'objectif est d'affiner la connaissance que nous avons de l'évolution Q avec des garanties. La première partie de la méthode est la Scan Matching que nous avons expliquée. La deuxième partie consiste à estimer au mieux l'évolution Q en utilisant différentes stratégies.

8.1 Problème de minimisation

Dans un premier temps nous avons voulu nous inspirer de la méthode utilisée par Yohan Breux qui vise à trouver l'évolution q du robot qui minimise la distance entre une mesure T2 et l'ensemble des mesures T1 qui lui sont associées. Il nous fallait trouver un équivalent de la distance de Mahalanobis en ensembliste. Nous avons utilisé la distance d'Hausdorff. En mathématiques, et plus précisément en géométrie, la distance de Hausdorff est un outil topologique qui mesure l'éloignement de deux sous-ensembles d'un espace métrique sous-jacent. J'ai créé un arbre de recherche basique permettant de trouver le sous-ensemble de Q qui minimise la somme :

$$\sum_{i=0}^n (\sum_{k=0}^l (distHausdorff(MT2i, MT1associe)))$$

Cette méthode est intéressante car elle permet de trouver un sous ensemble de l'évolution Q . Les résultats sont cependant non satisfaisants car le sous ensemble renvoyé n'inclut pas la solution Q exacte à chaque fois. Nous avons donc décidé d'abandonner cette méthode.

8.2 Contraction

Nous avons donc décidé de partir sur une méthode plus sûre et plus tournée intervalle. Pour utiliser cette méthode nous avons posé certaines hypothèses. En effet, comme on peut le voir sur la figure ci-dessous, nous ramenons (en vert) les mesures prises en T2 dans le repère du robot en T1 en appliquant la transformation Q estimée par l'IMU. Aussi de façon à utiliser des "box" qui permettent beaucoup de fonctionnalité avec TUBEX et IBEX on projette les mesures dans le repère cartésien.

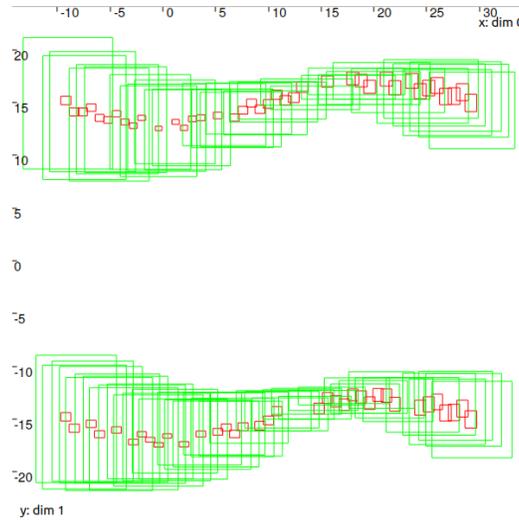


Figure 18: Toutes les mesures ramenées dans le repère du robot en T1

Du fait de l'incertitude de Q les mesures ramenées dans le référentiel du robot sont représentées par des boîtes plus grosses (l'incertitude se propage lors de la projection). Il faut donc trouver une méthode permettant de contracter les mesures vertes de façon à contracter l'évolution Q . Pour cela nous avons émis une hypothèse. Après réunion avec Yohan Breux concernant les différents capteurs utilisés, nous avons conclu que les capteurs utilisés permettent dans tous les cas de figures de donner le point le plus proche de la Karst par rapport au robot dans la zone balayée. Ainsi, lorsqu'on ramène les mesures dans un seul et même référentiel on peut considérer que la surface du mur a été identifiée à la première acquisition à T1. Donc pour l'acquisition à T2 on peut contracter l'ensemble incluant le mur jusqu'à la surface de celui-ci déjà identifiée. Nous avons considéré que les mesures rouges étant suffisamment proches les unes des autres, elles nous donnent une idée de l'emplacement du mur dans cette zone. Ainsi, nous pouvons considérer que la partie d'une boîte verte qui est plus proche du robot que le hull de ses boîtes associées est incohérente. Donc on peut utiliser les boîtes noires de la figure ci-dessous pour contracter les boîtes vertes.

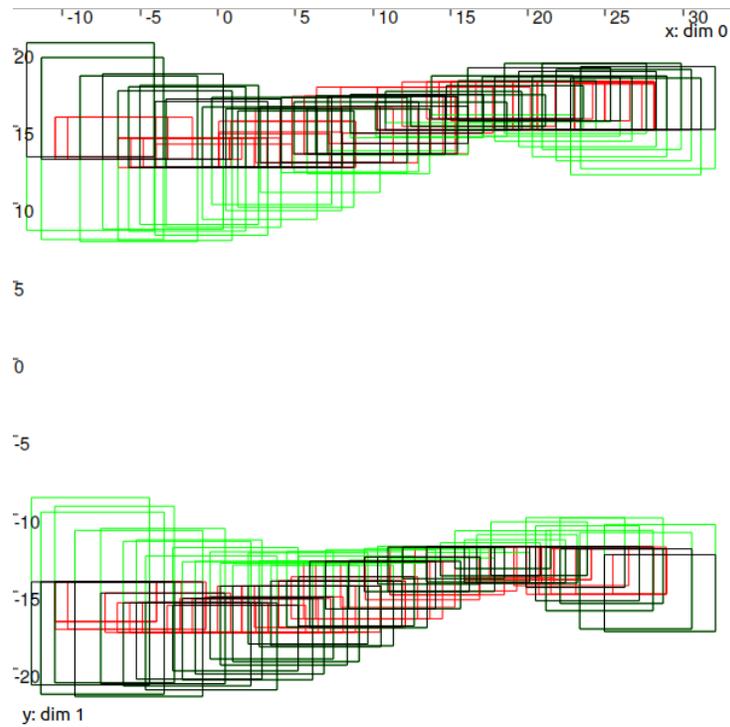


Figure 19: En noir, la partie de chaque boîte verte qui est cohérente.

On crée un réseau de contracteurs CN. Pour chaque mesure la contrainte mentionnée ci-dessus est imposée et ajoutée au réseau de contracteurs. A la fin de cette boucle, la contraction est effectuée.

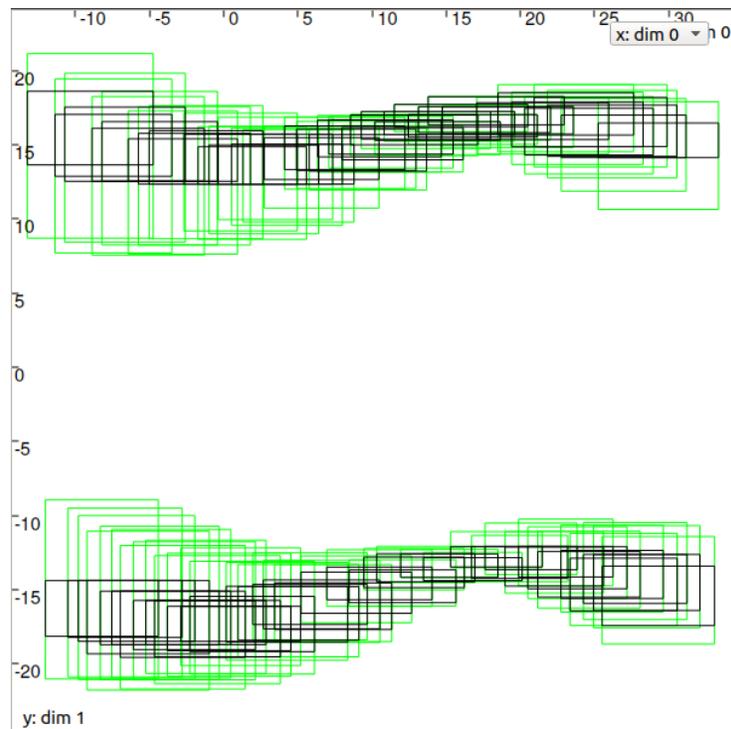


Figure 20: En noir le résultat de la contraction.

Le résultat de cette méthode est très positif puisque comme on peut le voir, l'incertitude caractérisant les boîtes vertes a très nettement réduit. De plus, on s'aperçoit que cette méthode permet de contracter Q . Une fois l'évolution Q contracter, elle inclut toujours la solution exacte. Donc on peut conclure sur le fait que cette méthode marche. Les résultats sont cependant insatisfaisants. La contraction de Q ne permet pas d'obtenir une estimation suffisamment fine. La principale raison pour laquelle cette méthode est trop peu précise est le wrapping effect qui est induit lors du passage des coordonnées polaires en cartésiens.

8.3 Système de coordonnées polaires

Lorsqu'on travaille avec des intervalles en 2D par exemple, on utilise des boîtes pour modéliser un ensemble. On raisonne sur ces boîtes pour trouver des solutions. Mais bien souvent une boîte n'est pas la meilleure des

formes pour englober un ensemble. Par exemple lorsqu'on veut représenter un ensemble de points inclut dans un cercle par une boite on induit un wrapping effect:

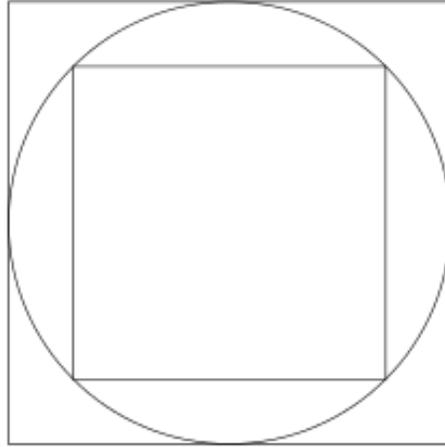


Figure 21: wrapping effect

Dans notre cas, les mesures ont initialement la forme d'une portion de disque (coordonnées polaires) et on wrap ces mesures dans des boites (rectangles). Le wrapping effect est donc très important. L'une des pistes envisagées a donc été de trouver un moyen d'amener les mesures de l'instant de T2 dans le repère du robot à T1 sans faire de projection et donc en restant avec le système de coordonnées polaires. Pour déterminer les coordonnées des mesures de T2 dans le repère T1 en restant en polaire il faut utiliser les équations suivantes.

$$\alpha r1 = \arctan\left(\frac{\widehat{Y}q + dr2 * \sin(\alpha r2 + \widehat{\Theta}q)}{\widehat{X}q + \alpha r2 * \cos(\alpha r2 + \widehat{\Theta}q)}\right)$$

$$dr1 = \frac{\widehat{X}q + dr2 * \cos(\alpha r2 + \widehat{\Theta}q)}{\cos(\alpha r1)}$$

avec $\alpha r1, dr1$ Les coordonnées d'une mesure T2 dans le repère T1. Le problème est que comme vous pouvez le voir, ces équations font intervenir

des occurrences multiples. Ce qui est difficile à gérer en intervalle. En effet, par exemple en intervalle $[i]-[i]$ n'est pas forcément égale à 0. La solution envisagée dans un premier temps a été d'ajouter ces équations au CN de façon à voir si ces contraintes pouvaient contracter d'avantage Q . Cette solution n'a pas été concluante. Cette approche reste selon moi la piste de recherche la plus intéressante dans l'idée d'améliorer cette technique.

8.4 Évolution envisagée

L'une des évolutions envisagées est de partir d'une technique de SLAM mise au point par Luc JAULIN intitulée Dig SLAM [7] with range only sensor. Cette méthode est très robuste surtout vis-à-vis du type de capteur que nous possédons. La méthode est de "creuser" l'ensemble initial afin de faire apparaître les surfaces (murs) qui composent l'environnement. L'avantage de cette technique est qu'elle permet véritablement de réaliser la carte de l'environnement et en même temps de se localiser. Aussi cette technique a été réalisée avec un type de capteur "range-only". Notre projet permet de bénéficier de capteurs plus élaborés. L'une des pistes de développement peut être d'adapter cette méthode avec des capteurs qui apportent plus d'informations.

9 Conclusion

9.1 Simulateur

Le simulateur développé correspond au diagramme des exigences. Il permettra de tester les techniques de SLAM réalisées par l'équipe. Le simulateur est très modulable avec la possibilité d'ajouter de nombreuses fonctionnalités. Il pourra donc évoluer dans le temps. D'un point de vue personnel, la création du simulateur a été l'occasion pour moi de découvrir différentes technologies essentielles au monde de la robotique.

9.2 ICP en analyse par intervalle

La méthode de contraction a donnée des résultats intéressant mais elle doit être retravaillée. En effet, le fait de rester en coordonnées polaires peut optimiser cette méthode. En travaillant sur ce sujet je me suis aperçu à quel point la recherche peut être difficile et en même temps très enrichissante. Ce sujet m'a permis de travailler avec des chercheurs spécialisés dans ce domaine qui m'ont beaucoup apporté. Cette expérience aura donc été extrêmement positive d'un point de vue personnel.

References

- [1] Sujet de thèse: Intervalles, systèmes dynamiques et trajectoires garanties pour la robotique d'exploration karstique.
- [2] Gilles Trombettoni Lionel Lapierre Simon Rohou. Intervalles, systèmes dynamiques et trajectoires garanties pour la robotique d'exploration karstique |. http://www.lirmm.fr/~trombetton/sujet_these_intervalles_systemes_dynamiques_karst.pdf.
- [3] Musa Morena Marcusso Manhães, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation, sep 2016.
- [4] LE TOLGUENEC Paul-Antoine. Karst world in gazebo |. https://github.com/Paul-antoineLeTolguenec/uuv_cave_world.
- [5] Probabilistic Scan Matching for Motion Estimation in Unstructured Environments | Luis Montesano Javier Minguez Luis Montano. http://webdiis.unizar.es/~montesan/papers/IR0505_SM.pdf.
- [6] Giorgio Grisetti Rainer Kummerle Cyril Stachniss Wolfram Burgard. A Tutorial on Graph-Based SLAM |. <http://ais.informatik.uni-freiburg.de/publications/papers/grisetti10titsmag.pdf>.
- [7] Luc JAULIN | Dig SLAM SLAM indoor. https://www.ensta-bretagne.fr/jaulin/paper_dig_slam.pdf.