



**ENSTA Bretagne**

Deuxième année Filière Ingénieur sous Statut Élève (FISE 2022)  
Spécialité Robotique Mobile

# Rapport de stage en entreprise

---

## Cartographie 2D et navigation

---

Travail réalisé par  
**Estelle ARRICAU**

Effectué au sein de  
**Kompäi Robotics**

Sous la direction de  
**Antoine TRIAUX - Ingénieur Robotique**

A Bidart (64)  
Du 26 avril au 27 août 2021

### **Encadrement**

Dr. Luc JAULIN  
Dr. Benoît ZERR  
Ing. Antoine TRIAUX

Référent ENSTA Bretagne  
Référent ENSTA Bretagne  
Tuteur de stage

# Résumé

L'activité principale chez Kompai Robotics consiste à produire un robot - le Kompai 3 - d'assistance aux personnes à mobilité réduite capable de naviguer en autonomie dans un environnement intérieur. Dans ce contexte, ma mission de stage s'inscrit dans le cadre de l'amélioration continue du robot. L'objectif est de changer le logiciel actuel de cartographie de l'environnement en respectant les contraintes logicielles et matérielles déjà existantes. Cette mission est née de la volonté de l'entreprise de centraliser tout leur savoir-faire dans la carte électronique principale du robot fonctionnant sous une distribution GNU/Linux (Ubuntu 18.04).

Toutes contraintes prises en compte, il s'agit donc d'abord de se renseigner théoriquement sur les différents algorithmes de SLAM disponibles sous ROS Melodic, pour ensuite les tester sur le robot, comparer leurs performances et enfin intégrer l'algorithme le plus efficace dans l'architecture logicielle du robot. Le but est alors d'obtenir une carte de l'environnement (dans lequel le robot se déplace) aussi bonne voire meilleure qu'avec le logiciel actuel, en termes de rapidité de génération, de consommation de ressources informatiques (CPU, RAM), et de qualité de la carte obtenue (fidélité à la réalité). L'utilisation d'amers pour une cartographie plus fiable de l'environnement reste à explorer.

**Mots clés :** cartographie 2D - SLAM - ROS - navigation

# Abstract

The main activity at Kompai Robotics is to produce a robot - the Kompai 3 - to assist people with reduced mobility, capable of navigating autonomously in an indoor environment. In this context, my internship mission is part of the continuous improvement of the robot. The objective is to change the current environment mapping software while respecting the existing software and hardware constraints. This mission was born from the company's desire to centralize all their know-how in the main electronic board of the robot running under a GNU/Linux distribution (Ubuntu 18.04).

Taking into account all the constraints, the first step was to learn theoretically about the different SLAM algorithms available under ROS Melodic, then to test them on the robot, compare their performances and finally integrate the most efficient algorithm into the robot's software architecture. The goal is then to obtain a map of the environment (in which the robot moves) as good or better than with the current software, in terms of speed of generation, consumption of computer resources (CPU, RAM), and quality of the map obtained (fidelity to reality). The use of landmarks for a more reliable mapping of the environment remains to be explored.

**Key words :** 2D mapping - SLAM - ROS - navigation

# Remerciements

Je tiens à remercier en premier lieu, l'entreprise Kompaï Robotics de m'avoir offert l'opportunité de réaliser mon stage de deuxième année de cursus ingénieur au sein de ses locaux.

Je remercie également Vincent DUPOURQUÉ, directeur général et président, de m'avoir chaleureusement accueillie au sein de son équipe.

Je remercie surtout Antoine TRIAUX, mon maître de stage et ingénieur roboticien, pour sa confiance, sa pédagogie, sa bienveillance à mon égard, et surtout pour m'avoir accompagnée et guidée tout au long de ce stage.

J'ai une pensée particulière pour Marc TRAONMILIN (ingénieur logiciel référent) pour sa patience et sa réactivité face à mes problèmes techniques, et Laetitia LERANDY (ingénieure cognitive) pour sa bonne humeur et son soutien.

Enfin, j'ai forcément une pensée en écrivant ces remerciements pour l'ensemble de l'équipe de Kompaï Robotics pour leur disponibilité, et leur entrain contagieux.

# Glossaire

**API** *Application Programming Interface* - Ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications (communication entre différents services ou produits).

**BDD** *Base De Données*

**CPU** *Central Processing Unit* - Processeur, unité de calcul d'une carte électronique.

**EHPAD** *Établissement d'Hébergement pour les Personnes Âgées Dépendantes*

**EKF** *Extended Kalman Filter* - Filtre de Kalman étendu permettant de déterminer la prédire et corriger la position du robot dans son environnement à chaque itération.

**FLR-SA** *Flèche Lumineuse de Rabattement – Suiveuse Autonome* - Flèche lumineuse de rabattement de voitures utilisée pour les travaux autoroutiers.

**HTTP** *Hypertext Transfer Protocol* - Protocole de communication informatique entre un client et un serveur pour le World Wide Web.

**IMU** *Inertial Measurement Unit* - Unité électronique de mesure inertielle du robot (accélération et gyroscope sur trois axes).

**KomNAV** *Kompaï NAVigation* - Librairie informatique pour la navigation autonome du robot Kompaï dans un environnement intérieur.

**LiDAR** *Light Detection And Ranging* - Télémètre laser.

**OS** *Operating System* - Système d'exploitation d'un ordinateur.

**POI** *Point Of Interest* - Point d'intérêt sur la carte.

**PURE** *Professional Universal Robotic Engine* - Contrôleur bas niveau temps réel du robot Kompaï.

**RAM** *Random Access Memory* - Mémoire vive d'un ordinateur.

**RFID** *Radio Frequency IDentification* - Identification d'un objet par ondes radio.

**ROM** *Random Access Memory* - Mémoire morte, permanente d'un ordinateur.

**ROS** *Robot Operating System* - *Middleware* robotique, plateforme de développement logicielle pour robot. Il s'agit d'un méta-système d'exploitation qui peut fonctionner sur un ou plusieurs ordinateurs et qui fournit plusieurs fonctionnalités : abstraction du matériel, contrôle des périphériques de bas niveau, mise en œuvre de fonctionnalités couramment utilisées, transmission de messages entre les processus et gestions des packages installés.

**RTLS** *Real Time Localization System* - Système de localisation en temps réel d'un robot.

**SLAM** *Simultaneous Localization And Mapping* - Algorithme réalisant simultanément la localisation d'un robot et la cartographie de son espace d'évolution.

**UDP** *User Datagram Protocol* - Protocole de communication Internet.

**URL** *Uniform Resource Locator* - Adresse informatique d'une page web.

**UWB** *Ultra WideBand* - Technologie radio capable d'utiliser un très faible niveau d'énergie pour des communications sans fil à courte portée et à large bande passante sur le spectre radioélectrique.

**WebRTC** *Web Real-Time Communication* - API HTTP pour faire circuler des données de façon bilatérale et exclusive entre deux navigateurs web.

# Table des matières

<b>Résumé</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Remerciements</b>	<b>iv</b>
<b>Glossaire</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Présentation de l'entreprise</b>	<b>2</b>
2.1 Historique . . . . .	2
2.2 Activités . . . . .	3
2.2.1 Robot Kompaï . . . . .	3
2.2.2 Robosoft Consulting . . . . .	5
2.3 Organisation . . . . .	6
<b>3 Analyse du problème</b>	<b>9</b>
3.1 Contexte économique et social . . . . .	9
3.2 Contexte technique . . . . .	9
3.2.1 Fonctionnement général du robot . . . . .	10
3.2.2 Le logiciel de cartographie . . . . .	14
3.3 Recueil du besoin . . . . .	15
3.3.1 Besoin et objectifs . . . . .	16
3.3.2 Exigences et contraintes . . . . .	18
<b>4 Réalisation</b>	<b>19</b>
4.1 Étude théorique des solutions de SLAM existantes . . . . .	19
4.2 <i>Benchmark</i> des SLAM retenus sous ROS . . . . .	19
4.2.1 Temps de traitement . . . . .	20
4.2.2 Consommation des ressources informatiques . . . . .	21
4.2.3 Qualité des cartes obtenues . . . . .	22
4.3 Tâches annexes . . . . .	28
<b>5 Gestion de projet</b>	<b>30</b>
5.1 Organisation . . . . .	30
5.2 Problèmes rencontrés . . . . .	31
<b>6 Conclusion</b>	<b>32</b>
<b>Bibliographie</b>	<b>32</b>

<b>7</b>	<b>Annexes</b>	<b>34</b>
7.1	Robosoft Consulting . . . . .	34
7.1.1	La Poste . . . . .	34
7.1.2	APRR . . . . .	34
7.1.3	Bio-UV . . . . .	35
7.2	Contexte économique et social . . . . .	37
7.2.1	Étude de marché . . . . .	37
7.2.2	Enjeux de développement durable . . . . .	39
7.2.3	Application de robotique sociale et médicale . . . . .	40
7.3	Étude théorique des solutions de SLAM existantes . . . . .	41
7.3.1	La méthode SLAM . . . . .	41
7.3.1.1	Présentation . . . . .	41
7.3.1.2	Principe . . . . .	41
7.3.2	Les différents SLAM . . . . .	43
7.3.2.1	Filtrage Bayésien . . . . .	43
7.3.2.2	Optimisation de graphe . . . . .	44
7.3.2.3	Vision et intelligence artificielle . . . . .	44
7.3.3	Comparaison théorique des différents SLAM 2D sous ROS . . . . .	45
7.3.3.1	CRSM SLAM . . . . .	45
7.3.3.2	TinySLAM et CoreSLAM . . . . .	45
7.3.3.3	LagoSLAM . . . . .	45
7.3.3.4	HectorSLAM . . . . .	46
7.3.3.5	KartoSLAM . . . . .	46
7.3.3.6	GMapping . . . . .	47
7.3.3.7	Cartographer . . . . .	47
7.3.3.8	Bilan . . . . .	48
7.4	Modifications du logiciel de cartographie <b>Slam2d</b> . . . . .	49
7.5	Veille technologique sur le système RTLS de Terabee . . . . .	51
7.6	Organigrammes logiciels, matériels et réseaux . . . . .	53
7.7	ROS node graph . . . . .	59

# Table des figures

2.1	Robot Kompaï . . . . .	3
2.2	Évolution de l'apparence du Kompaï depuis sa création chez ROBOSOFT . . . . .	4
2.3	Composants du robot Kompaï . . . . .	4
2.4	Organigramme de l'équipe de Kompaï Robotics . . . . .	6
2.5	Organisation des différents pôles d'activité chez Kompaï Robotics . . . . .	8
3.1	Schéma logiciel du robot Kompaï 3 . . . . .	10
3.2	Visualisation du <i>dashboard</i> onglet <b>Navigation</b> . . . . .	11
3.3	Fonctionnement en arborescence du logiciel PURE . . . . .	12
3.4	Schéma fonctionnel du robot Kompaï 3 . . . . .	13
3.5	Interface utilisateur du logiciel de cartographie <b>Slam2d</b> . . . . .	15
3.6	Les étapes d'amélioration continue en cours sur le logiciel de cartographie . . . . .	16
4.1	Visualisation de la fenêtre RViz lors de la cartographie . . . . .	20
4.2	Consommation moyenne de RAM en Mo pendant la cartographie . . . . .	21
4.3	Consommation moyenne de CPU en % pendant la cartographie . . . . .	21
4.4	Cartes des locaux de Kompaï Robotics obtenues avec les différents SLAM . . . . .	24
4.5	Cartes du premier étage obtenues avec les différents SLAM . . . . .	25
4.6	Cartes de l'ESTIA obtenues avec les différents SLAM . . . . .	26
4.7	Cartes de l'EPHAD d'Arcangues obtenues avec les différents SLAM . . . . .	27
4.8	Nouvelle IHM de cartographie et de création de rondes . . . . .	28
5.1	Planning prévisionnel pour ma mission de stage . . . . .	30
5.2	Planning effectif pour ma mission de stage . . . . .	30
7.1	Chariot suiveur La POSTE (modélisation à gauche et prototype rendu à droite) . . . . .	34
7.2	Robot suiveur pour signalisation de travaux sur autoroute . . . . .	35
7.3	Robot Kompaï portant le module de désinfection Bio-UV . . . . .	36
7.4	Évolution du marché de la robotique d'assistance entre 2018 et 2025 . . . . .	37
7.5	Robot Nao qui anime une séance de sport dans un EHPAD . . . . .	38
7.6	Robot Cutii de la société CareClever . . . . .	39
7.7	Organigramme du processus SLAM . . . . .	42
7.8	Classification des différentes méthodes SLAM . . . . .	43
7.9	Comparaison des différentes méthodes SLAM 2D sous ROS . . . . .	48
7.10	Nouvel onglet <i>Default</i> dans le logiciel <b>Slam2d</b> . . . . .	50
7.11	Nouvel onglet <i>Advanced</i> dans le logiciel <b>Slam2d</b> . . . . .	51
7.12	Communications réseaux possibles sur le robot Kompaï . . . . .	53
7.13	Architecture matérielle globale du robot Kompaï . . . . .	54
7.14	Architecture matérielle globale détaillée du robot Kompaï . . . . .	55

---

7.15	Architecture matérielle détaillée des moteurs, du panneau de configuration et de l'alimentation du Kompai . . . . .	56
7.16	Architecture matérielle détaillée des capteurs, des télécommunications et des actions utilisateur du Kompai . . . . .	57
7.17	Architecture matérielle détaillée de la tablette et de la Raspberry Pi du Kompai .	58
7.18	<i>ROS node graph</i> après intégration des noeuds de détection d'obstacles et de cartographie avec PURE . . . . .	59

# 1 Introduction

L'entreprise d'accueil possède deux activités. La première, est l'industrialisation du robot d'assistance Kompai destiné aux résidents d'EHPAD et leurs aidants. La deuxième activité consiste à conseiller, élaborer des preuves de concept et accompagner les clients dans leur transition robotique. Ma mission de stage s'inscrit dans un contexte d'amélioration continue du robot Kompai : le but est de remplacer le logiciel propriétaire actuel de cartographie de l'environnement par un logiciel libre équivalent en termes de résultats mais fonctionnant sous ROS<sup>1</sup>.

La mission principale de ce stage consiste donc à évaluer différents algorithmes de SLAM<sup>2</sup> intégrables sous ROS Melodic. En effet, le robot principal produit par l'entreprise - le Kompai 3<sup>3</sup> - nécessite de pouvoir naviguer en autonomie dans l'environnement intérieur dans lequel il évolue. Cette navigation autonome requiert donc au préalable une cartographie du lieu qui est réalisée grâce à la combinaison de données laser 2D et de données odométriques. Le microcontrôleur principal du robot (JETSON Nvidia TX2) fonctionnant sous une distribution GNU/Linux (Ubuntu 18.04), l'entreprise a récemment choisi d'intégrer ROS pour faire communiquer les capteurs et les actionneurs du robot entre eux.

Il s'agit donc d'abord de se renseigner théoriquement sur les différents algorithmes de SLAM disponibles sous ROS, pour ensuite les tester, comparer leurs performances et enfin intégrer l'algorithme le plus efficient dans l'architecture logicielle du robot.

De petites missions secondaires me sont aussi attribuées afin de participer à l'amélioration continue du robot Kompai 3. Ces missions peuvent être de la veille technologique, des modifications d'interface utilisateur ou encore la réalisation de schémas de fonctionnement du robot.

---

1. Robot Operating System
2. Simultaneous Localization And Mapping
3. Numéro de version du robot Kompai actuel

## 2 Présentation de l'entreprise

**Kompaï Robotics** est une start-up de robotique mobile installée à Bidart, dans le Pays Basque français (64), et créée en 2016 par reprise d'une partie des actifs de la précédente société **Robosoft**.

### 2.1 Historique

**1985** Création de l'entreprise **Robosoft** par Vincent DUPOURQUÉ et qui a pour vocation de robotiser les services, à l'image de la production industrielle qui s'est robotisée dans les années 60.

**2008** Création de **Robulab**, la première base de robotique mobile, et de **Roburide**, un robot de transport de personnes en réponse à un projet européen **CityMobil** de modernisation de la mobilité citadine. Ces deux robots sont équipés du logiciel de contrôle bas niveau **PURE**<sup>1</sup> développé depuis la création de l'entreprise.

**2012** Sortie du **Kompaï 1**, un robot compagnon destiné à assister les personnes à domicile.

**2015** Sortie du **Kompaï 2**, une évolution de son prédécesseur, et participation à un projet européen de robot d'assistance aux personnes âgées. Un déambulateur automatique est conçu par Robosoft sur ce projet, ce qui donne l'idée de la mise en place de barres de déambulation sur le **Robulab**. C'est ainsi que Robosoft transfère peu à peu son activité principale vers l'aide à la personne.

**2016** Malheureusement, l'entreprise Robosoft ne connaît pas un succès suffisant et est placée en redressement judiciaire. En parallèle, l'entreprise Kompaï Robotics est créée et présidée par Vincent DUPOURQUÉ, un ancien directeur de Robosoft.

**2017** Conversion du redressement en liquidation judiciaire et mise en place d'un plan de cessation partielle d'activité pour Robosoft. Kompaï Robotics rembauche alors une partie des effectifs de Robosoft.

**2019** Sortie du **Kompaï 3** sous le nom de la nouvelle entreprise Kompaï Robotics.

**2020** Obtention d'une levée de fonds d'un montant de 200 000 euros à condition de produire au moins 20 robots **Kompaï** et d'embaucher 20% de personnel en plus d'ici 2022.

---

1. Professional Universal Robotic Engine

## 2.2 Activités

Kompaï Robotics développe deux activités :

- Proposer des solutions robotiques d'assistance aux personnes fragilisées et leurs aidants, à destination des établissements de santé mais aussi pour le maintien à domicile. Il s'agit de la gamme du robot **Kompaï** qui est en phase de tests et de mise au point dans l'EHPAD d'Arcangues avec lequel l'entreprise travaille en collaboration. Ce robot est d'ailleurs en phase d'industrialisation. À l'heure actuelle, il est entièrement fabriqué par l'entreprise : de sa conception à son intégration sur site client, en passant par l'assemblage mécanique et électronique, ou encore, sa paramétrisation et sa mise au point logicielles.
- **Robosoft Consulting**, activité d'ingénierie et de conseil, qui a pour objectif d'accompagner dans leur transition robotique ses clients quel que soit leur domaine d'activité. Pour cela, Kompaï Robotics développe des preuves de concept, élabore des scénarios d'usage, participe à l'industrialisation et au déploiement des solutions robotiques de ses clients.

### 2.2.1 Robot Kompaï



FIGURE 2.1 – Robot Kompaï

Kompaï (voir Figure 2.1), fruit de nombreuses années de recherches (voir Figure 2.2), est un robot qui se veut utile au personnel soignant mais aussi aux patients des établissements de santé. L'intérêt est de le programmer toute la journée pour différentes fonctions, afin de décharger les plus occupés, et occuper les plus isolés.

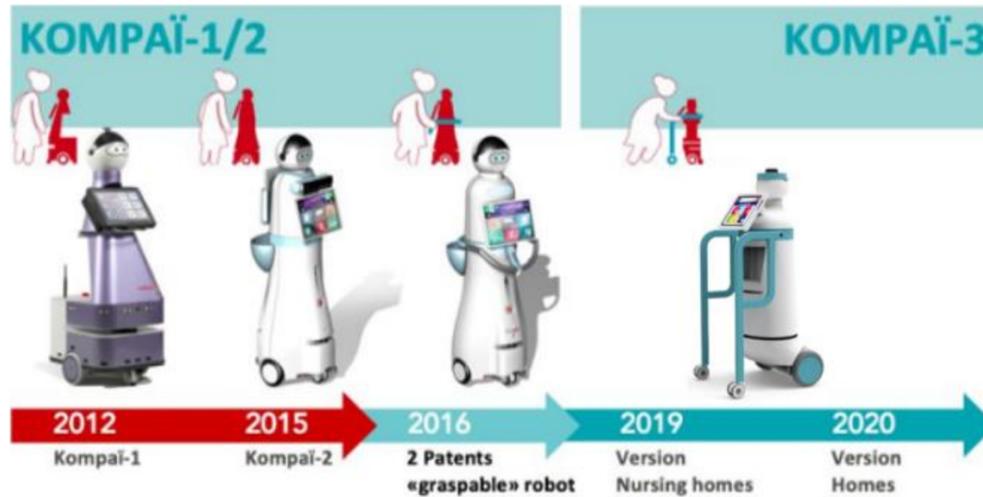


FIGURE 2.2 – Évolution de l'apparence du Kompaï depuis sa création chez ROBOSOFT

Son apparence générale est issue de plusieurs *focus group*, qui ont conduit à définir les dimensions et les limites de l'aspect humanoïde afin d'obtenir un robot qui inspire confiance. Ainsi, ce dernier est de la taille d'un enfant et totalement asexué. Il ne fait que suggérer une dimension humanoïde mais sans ambiguïté quant à la volonté de se différencier des humains. Si les premières versions étaient des robots mobiles assez classiques, l'évolution majeure s'est faite en 2019 avec l'apparition de la barre d'assistance à la marche, à la demande des utilisateurs. Ainsi, Kompaï aide à marcher, synonyme d'une plus grande autonomie pour les utilisateurs. Cette solution s'appuie sur un mécanisme breveté, une barre rotative, qui agit comme un volant de voiture : le robot prend la direction donnée par la barre (voir Figure 2.3).



FIGURE 2.3 – Composants du robot Kompaï

Aujourd'hui les principaux composants (voir Figure 2.3) du Kompaï sont :

- Le laser rotatif sur la tête, assurant la localisation dans une carte pré-enregistrée ;

- Les yeux animés, donnant des indications sur l'état du robot ;
- La caméra 3D, utilisée entre autres pour la détection des obstacles ;
- La caméra « grand-angle », utilisée pour la prise de contrôle du robot à distance ;
- Le laser bas, complète la caméra 3D pour la gestion des obstacles ;
- La tablette, permettant de lancer les applications ;
- Les barres de soutien, pour l'aide à la marche ;
- Le laser permettant d'adapter la vitesse du robot à celle des utilisateurs.

Depuis sa création, il n'y a pas que l'apparence du robot et ses composants qui ont évolués. De nombreuses preuves de concept ont été testées afin de définir au mieux les fonctionnalités adaptées aux établissements de santé et plus particulièrement aux EHPAD. Parmi ces fonctionnalités, les plus attendues par les EHPADs sont :

- **La surveillance** - De jour comme de nuit, le robot peut, sur un parcours prédéfini, détecter une anomalie (une chute, une personne sortie de sa chambre, etc.) et envoyer une alerte aux personnes référentes.
- **Les tournées** - Le robot peut faire des annonces (date, heure, activités de la journée, etc.) ou diffuser de la musique tout en ayant un parcours prédéfini dans l'établissement.
- **L'aide à la marche** - Grâce à ses barres intégrées, Kompai aide les patients à marcher, soit pour faire de l'exercice, soit pour se déplacer en parcours autonome ou guidé.
- **Le divertissement** - La tablette du Kompai permet de divertir les résidents individuellement ou collectivement et de maintenir le lien avec la famille (Skype, jeux cognitifs, karaoké, horoscope, vidéo, musique, contes, etc.).

L'ensemble de ces actions représentent six à huit heures de travail quotidien pour le robot, et ceci 365 jours par an. Les tâches réalisées par le robot sont essentiellement répétitives et à faible valeur ajoutée. Elles permettent ainsi de dégager du temps au personnel soignant pour consacrer plus de temps au contact des résidents.

Kompai existe aussi en version R&D. Il s'agit d'une version modulaire destinée aux universités et à tous ceux qui souhaitent personnaliser leur robot, que ce soit sur le plan matériel ou logiciel :

- Au niveau matériel, il est possible de substituer la barre d'assistance à la marche par un autre support permettant entre autres de mettre en place diverses missions en rapport avec le transport autonome de produits ;
- Au niveau logiciel, il est possible de développer n'importe quelle application embarquée ou débarquée, en s'appuyant sur les requêtes HTTP ou les accès URL développer par Kompai Robotics.

### 2.2.2 Robosoft Consulting

Afin de mieux comprendre l'activité côté Robosoft Consulting, une présentation de trois robots réalisés pour des appels d'offre est disponible en Annexe 7.1.

## 2.3 Organisation

L'entreprise compte en 2021 12 collaborateurs, et possède un capital social de 213 500 euros en 2020.

L'équipe de l'entreprise est hiérarchisée comme indiqué sur la Figure 2.4.

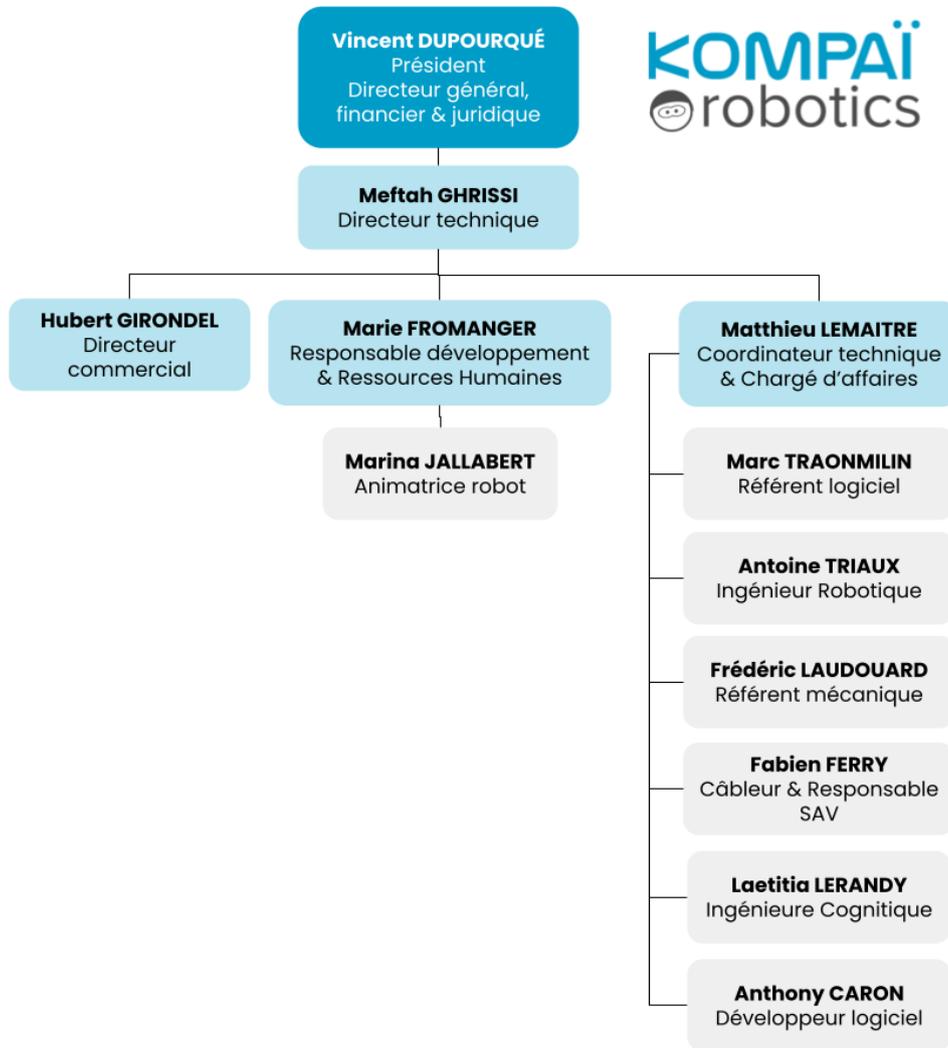


FIGURE 2.4 – Organigramme de l'équipe de Kompaï Robotics

L'organisation de l'équipe (voir Figure 2.5) est la suivante :

— L'Atelier

- modélise l'ensemble des pièces des robots et conçoit les pièces mécaniques personnalisées,
- réalise les calculs de faisabilité d'une évolution sur les robots (encombrement, efforts mécaniques, centre de gravité, etc.),

- passe et reçoit les commandes de pièces mécaniques et électroniques,
- assemble toutes les pièces du robot Kompai et le câble électroniquement.

L'atelier est aussi responsable du service client. C'est donc lui qui réalise les réparations et remplacements de pièces nécessaires sur les robots.

— Le pôle **R&D**

- développe logiquement toutes les fonctionnalités intergicielles et bas niveau du robot,
- améliore continuellement les fonctionnalités du robot Kompai,
- lance logiquement le robot Kompai une fois assemblé et câblé par l'atelier,
- participe activement au service client en déboguant ce qui ne fonctionne pas.

— Le pôle **Développement Logiciel**

- recueille le besoin client en termes d'interfaces utilisateur,
- implémente les interfaces utilisateur à la fois côté *front-end* et côté *back-end*.

— Le pôle **Commercial**

- promeut le robot Kompai sur des événements tels que des salons,
- débusque de nouveaux clients potentiels,
- gère les ressources humaines de l'entreprise.

— Le pôle **Accompagnement du Client**

- s'occupe d'intégrer dans les établissements cibles le robot Kompai auprès du personnel soignant et des patients,
- réalise des formations d'utilisation du robot Kompai,
- suit le client dans sa transition robotique en milieu médical.

— Le pôle **Finance & Administration**

- gère les contrats avec les clients, les réponses à appel d'offre,
- gère les comptes financiers, les aspects juridiques, et les investissements de l'entreprise,
- donne son accord final lors de la prise d'une décision pour l'entreprise.

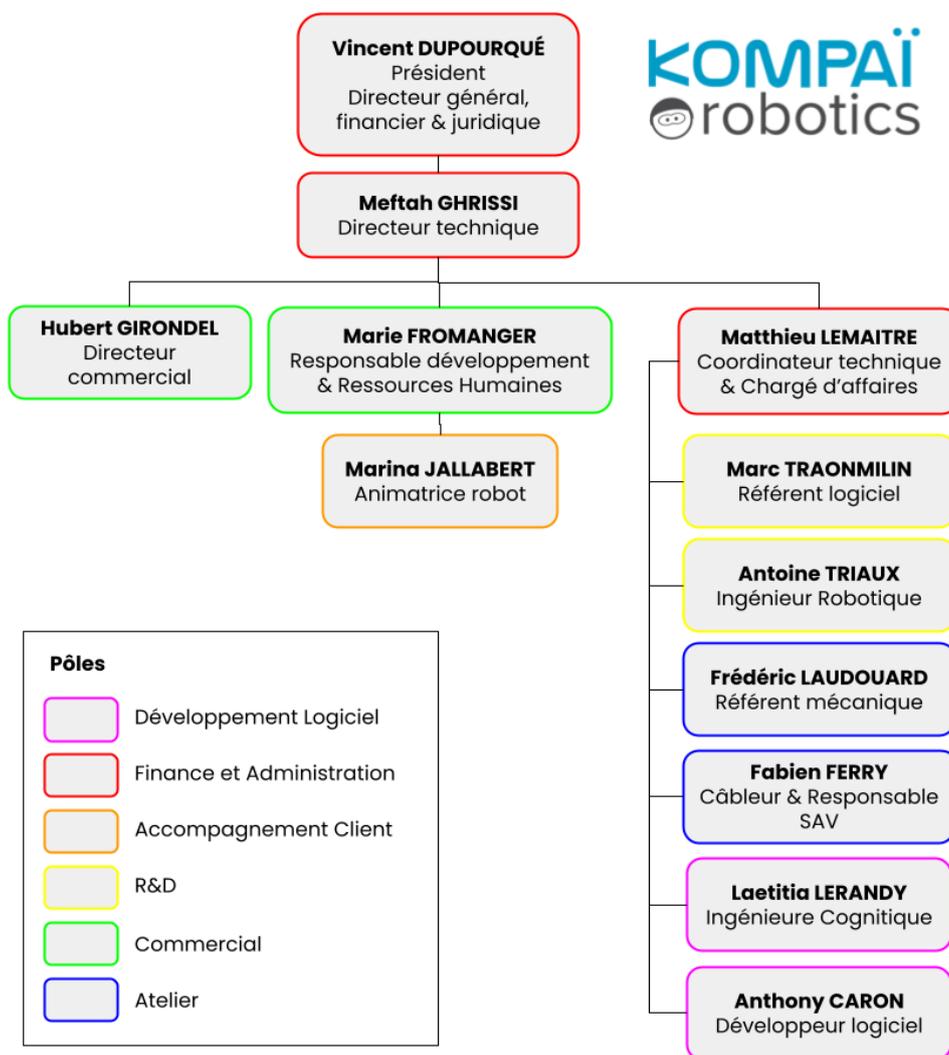


FIGURE 2.5 – Organisation des différents pôles d'activité chez Kompaï Robotics

## **3 Analyse du problème**

Cette analyse du problème s'attache à appréhender ma mission de stage, notamment pour clairement identifier les objectifs et les enjeux qu'elle représente, les acteurs impliqués, mon champ d'action dans un environnement technique imposé.

La mission de stage sera donc d'abord replacée dans un contexte global avant de se concentrer sur son contexte local.

### **3.1 Contexte économique et social**

Avant de s'intéresser aux objectifs, il est primordial de prendre en compte le contexte de ma mission : elle s'inscrit dans le domaine de la robotique mobile sociale pour le domaine médical.

Aujourd'hui, le robot Kompaï répond à une problématique économique et sociale qui sera de plus en plus pesante sur le moyen à long terme sur le monde occidental. C'est une application de robotique médicale et sociale qui a de l'avenir mais qui est encore trop en avance par rapport aux priorités actuelles des pays occidentaux, notamment en France. Une étude de marché considérant les enjeux de développement durable et les enjeux de la robotique sociale et médiale est disponible en Annexe 7.2.

### **3.2 Contexte technique**

Il s'agit de décrire l'environnement technique dans lequel ma mission de stage s'inscrit pour en extraire les tenants et les aboutissants.

### 3.2.1 Fonctionnement général du robot

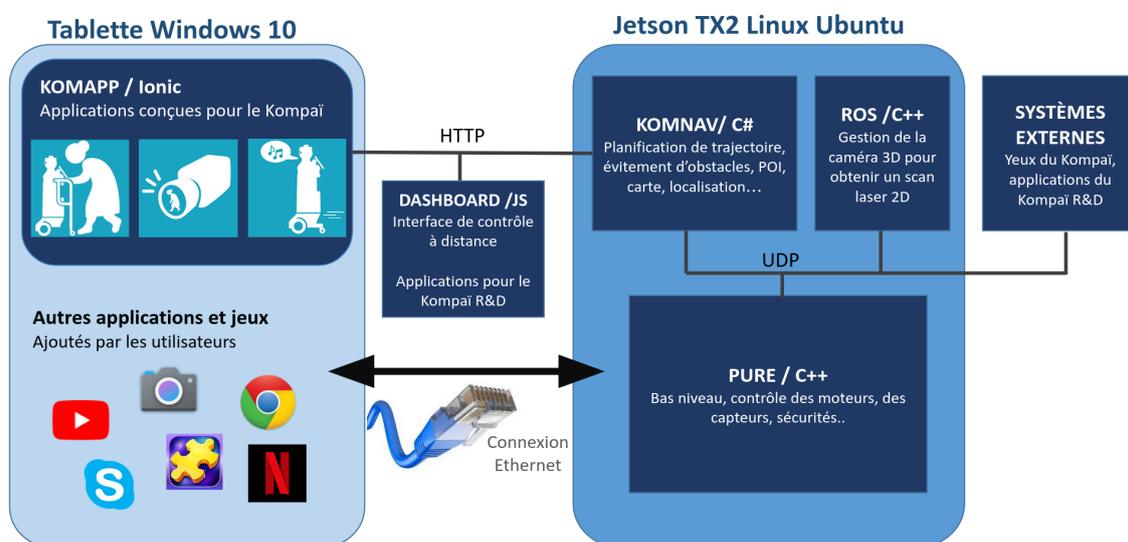
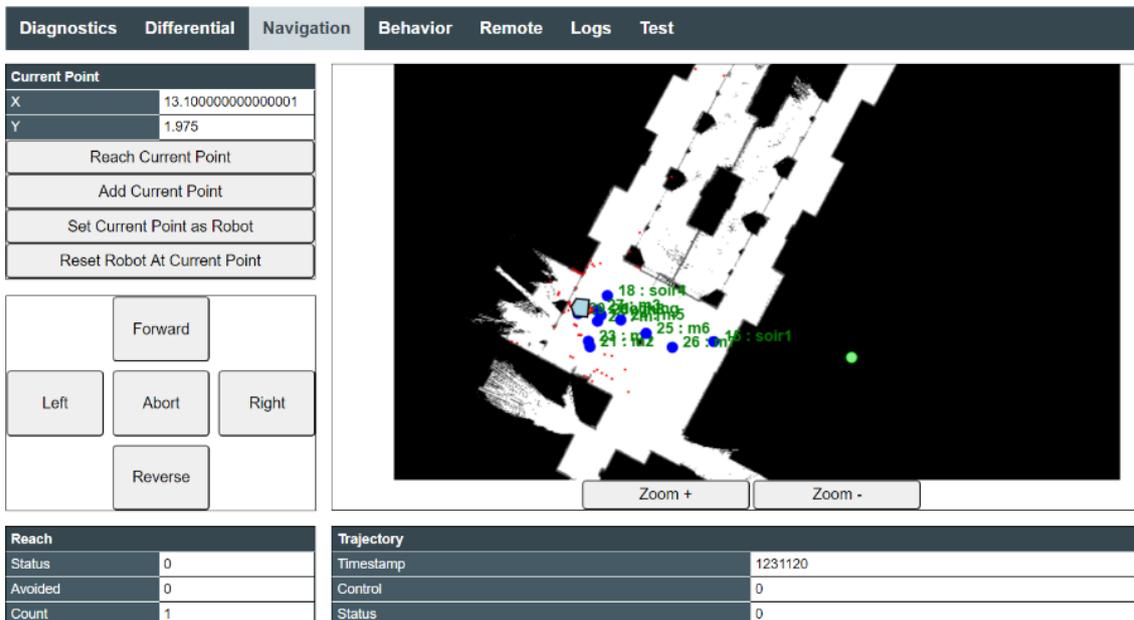


FIGURE 3.1 – Schéma logique du robot Kompaï 3

Le robot Kompaï possède différentes briques logicielles (voir Figure 3.1) :

- Les applications destinées à l'utilisateur final sont toutes sur la tablette Windows. Elles comprennent notamment les **KomAPP** qui sont les applications spécifiquement développées par Kompaï Robotics pour l'aide à la marche, la surveillance des résidents en EPHAD, et les tournées de divertissement.
- Le *dashboard*, qui est hébergé sur la JETSON mais aussi accessible depuis la tablette Windows, permet de visualiser toutes les variables d'état du robot en temps réel. C'est également depuis cette interface, qu'à l'heure actuelle, les rondes sont créées (voir Figure 3.2), et qu'elles sont lancées. D'autres fonctionnalités telles que diagnostiquer des erreurs (l'aide au débogage), donner des ordres de déplacement au robot (instruction du type *go to point*) sont possibles.
- Le *middleware ROS* sert à traiter les données de la caméra 3D : il convertit le nuage de points 3D de la caméra en un scan laser 2D par projection sur le plan horizontal. Dans un objectif d'amélioration continue, toutes les données recueillies par PURE sont petit à petit transférées sur ROS pour pouvoir utiliser de plus en plus d'applications disponibles sous ROS.
- La librairie **KomNAV** est dédiée à la navigation autonome du robot Kompaï dans un environnement intérieur. Ces rôles sont nombreux :

- Planification de trajectoire grâce à un algorithme A\* avec une heuristique distance euclidienne sur une carte PNG. La librairie génère une liste de points de passage avec des instructions d'approche (vitesse maximale sur le segment, rampes d'accélération et de décélération) ;
- Localisation du robot à l'aide du laser 2D, de l'odométrie et de la carte de l'environnement. Les informations capteurs sont communiquées par PURE. KomNAV génère le modèle d'observation (filtre particulaire) et l'envoi à PURE qui exécute un filtre de Kalman pour mettre à jour les données de position du robot ;
- Stockage de données avec deux Bases De Données (BDD) : une BDD de paramètres (cartes réalisées, POI<sup>1</sup>, rondes), une BDD avec les données de fonctionnement (logs, états des variables) ;
- Gestion de l'arbre de comportement du robot : détermination du comportement à adopter en fonction de la situation (les différentes stratégies peuvent être testées en séquentiel, en parallèle, ou par ordre de priorité) ;
- Communication avec les interfaces utilisateur sur la tablette à travers son API<sup>2</sup> HTTP comme par exemple l'envoi d'un flux vidéo en temps réel pour la surveillance des résidents en utilisant WebRTC<sup>3</sup> ;
- Hébergement du dashboard pour la visualisation des données de fonctionnement en temps réel comme mentionné précédemment ;
- Communication UDP<sup>4</sup> avec la Raspberry Pi 3 et avec le logiciel PURE.

FIGURE 3.2 – Visualisation du *dashboard* onglet **Navigation**

1. Point Of Interest
2. Application Programming Interface
3. API HTTP pour faire circuler des données de façon bilatérale et exclusive entre deux navigateurs web
4. User Datagram Protocol

- Le logiciel **PURE** est le contrôleur bas niveau temps réel du robot Kompaï. Ces rôles sont nombreux :
  - Contrôle de vitesse ;
  - Contrôle de direction ;
  - Anti-collision ;
  - Localisation du robot : données d'odométrie ou GPS (quand il y a) et intégration des données de localisation provenant de **KomNAV** pour le filtre de Kalman ;
  - Suivi de trajectoire planifiée par **KomNAV** ;
  - Gestion des pilotes des capteurs (récupération de données) et actionneurs (envoi de commandes).

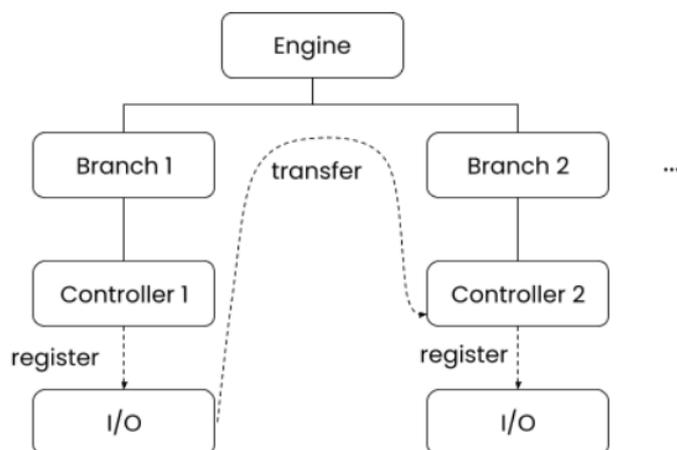


FIGURE 3.3 – Fonctionnement en arborescence du logiciel PURE

PURE fonctionne en arborescence pour paralléliser les tâches (voir Figure 3.3) : au démarrage du robot, PURE charge tous ses fichiers de configuration et lance le moteur de PURE (*Engine*). Ce moteur est à la racine de l'arborescence et gère le transfert des variables entrée/sortie aux branches (*Branch*).

Chaque branche gère un *thread*, c'est-à-dire un fil d'exécution pour une tâche spécifique. C'est elles qui permettent la prallélisation. Chaque branche possède un contrôleur (*Controller*) qui représente une tâche synchrone avec l'horloge de PURE.

Un contrôleur peut enregistrer des variables d'entrée/sortie pour être notifié de ses changements de valeur. À la place de ces contrôleurs, les gestionnaires (*Handler*) peuvent être mis en place. La différence est que le gestionnaire représente une tâche asynchrone pour la lecture de données capteurs (qui ont leur propre horloge).

Il communique à travers le protocole CANopen avec les capteurs et les différentes briques logicielles de PURE. Ce protocole gère les variateurs, les encodeurs, les variables entrées/sorties communiquées aux autres branches.

L'architecture macroscopique et les communications entre les différentes briques matérielles ou logicielles sont récapitulées dans la Figure 3.4.

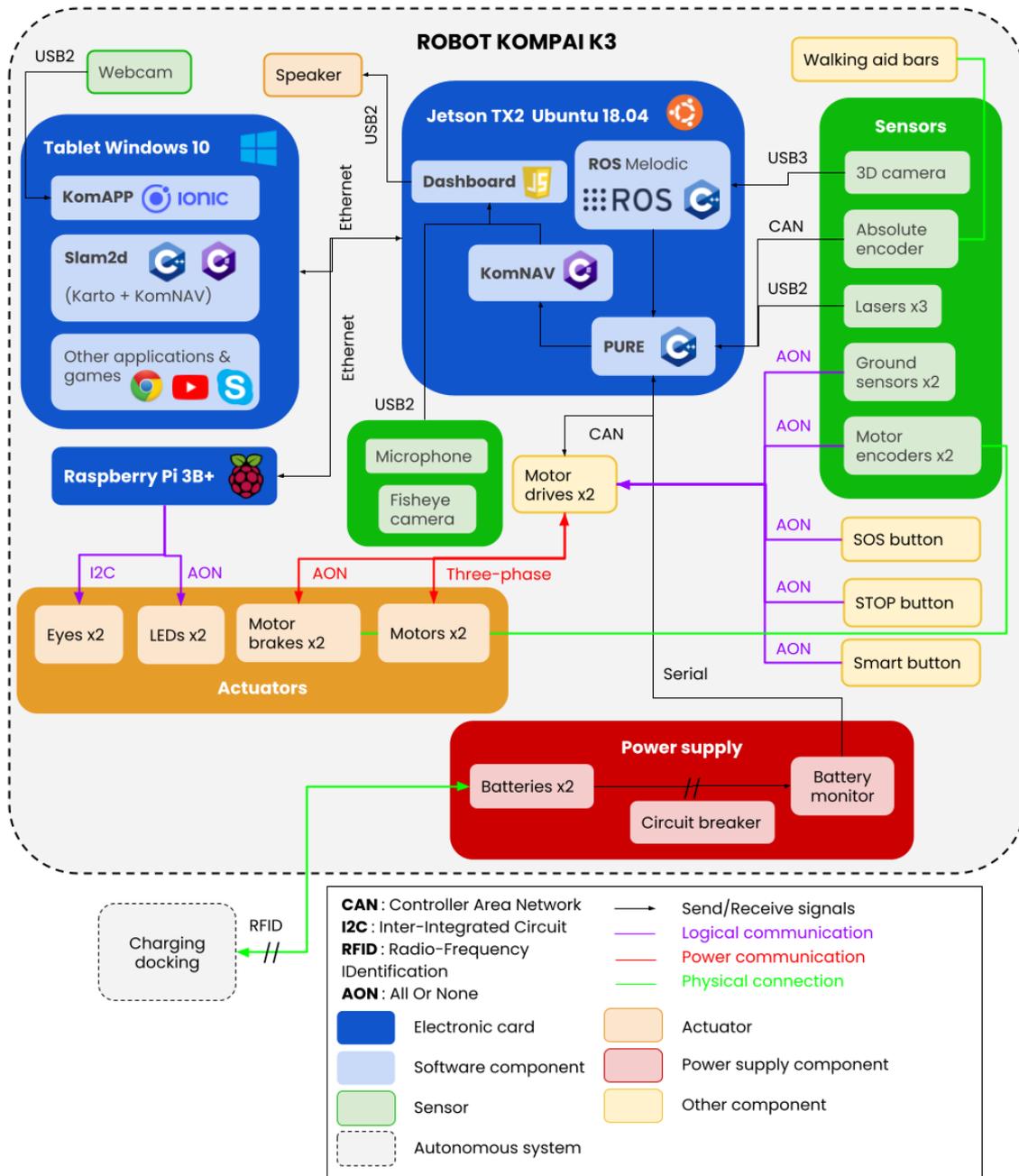


FIGURE 3.4 – Schéma fonctionnel du robot Kompai 3

De façon macroscopique, chaque élément a un rôle dans le fonctionnement du robot Kompai :

- La webcam est utilisée pour la surveillance des résidents ;
- **KomAPP** sont les applications utilisateur à destination des résidents ou du personnel

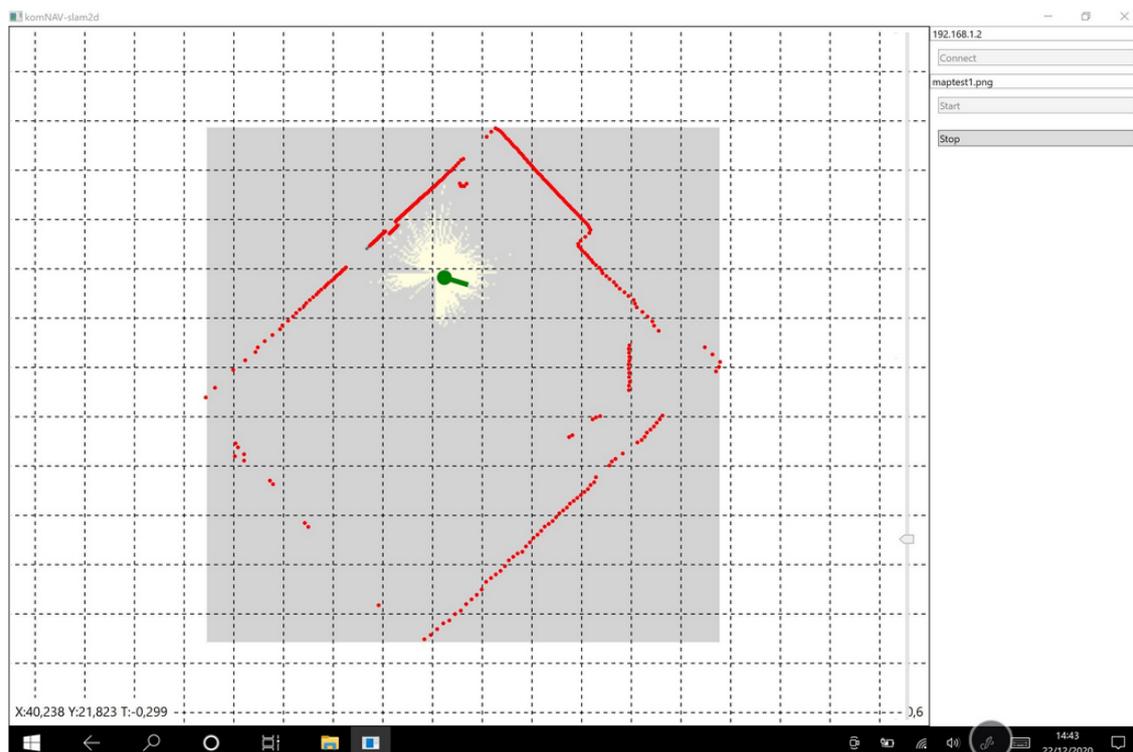
- médical ;
- **Slam2d** est le logiciel de cartographie de l'environnement ;
- Le microphone et le haut-parleur servent à faciliter les opérations de maintenance du robot à distance. L'opérateur communique grâce à ce microphone avec un employé Kompaï Robotics à distance ;
- La caméra *fisheye* sert à avoir un retour visuel sur le *dashboard* lors du contrôle du déplacement du robot à distance ;
- Les yeux et LEDs du robot indiquent l'état du robot (perdu, en déplacement vers un POI, etc.) ;
- La station de recharge reconnaît le robot grâce à une puce RFID<sup>5</sup> et envoie du courant à travers ses deux plots de charge lorsqu'ils sont correctement positionnés ;
- Les barres d'aide à la marche servent à diriger le robot, l'encodeur absolu détermine la direction donnée par les barres ;
- La caméra 3D détecte les obstacles dans l'environnement ;
- Le laser du haut est utilisé pour la cartographie et la localisation ;
- Le laser du milieu détecte le mouvement des genoux de la personne qui utilise l'assistance à la marche ;
- Le laser du bas sert également à la détection des obstacles ;
- Le bouton SOS appelle le personnel médical en cas d'urgence ;
- Le bouton STOP arrête les moteurs du robot ;
- Le bouton malin déverrouille les freins pour pouvoir déplacer manuellement le robot ;
- Le moniteur de batterie affiche des informations utiles telles que le niveau de batterie ou le courant qui circule dans le circuit de charge ;
- Le sectionneur coupe toute alimentation électrique au robot.

### 3.2.2 Le logiciel de cartographie

Initialement, le logiciel existant de cartographie **Slam2d** se trouve sur la tablette Windows et se présente comme sur la Figure 3.5.

---

5. Radio Frequency IDentification

FIGURE 3.5 – Interface utilisateur du logiciel de cartographie **Slam2d**

Pour l'utiliser, il faut :

1. Renseigner l'adresse IP de la JETSON pour récupérer les données laser (en jaune et rouge sur la Figure 3.5) ;
2. Se connecter à la JETSON ;
3. Renseigner le nom de la carte avec l'extension ".png" obligatoire ;
4. Cliquer sur le bouton *Start* pour commencer la cartographie ;
5. Cliquer sur le bouton *Stop* pour arrêter la cartographie. La carte est alors enregistrée dans un dossier prédéfini.

**Remarque** Dans son algorithme, il repose sur le SLAM Karto avec un optimeur plus performant (voir Annexe 7.3 pour un détail sur les méthodes SLAM).

### 3.3 Recueil du besoin

Il s'agit de décrire le besoin au plus proche de la réalité utilisateur afin de cerner au mieux les enjeux de la mission, et donc d'y répondre de façon adéquate.

### 3.3.1 Besoin et objectifs

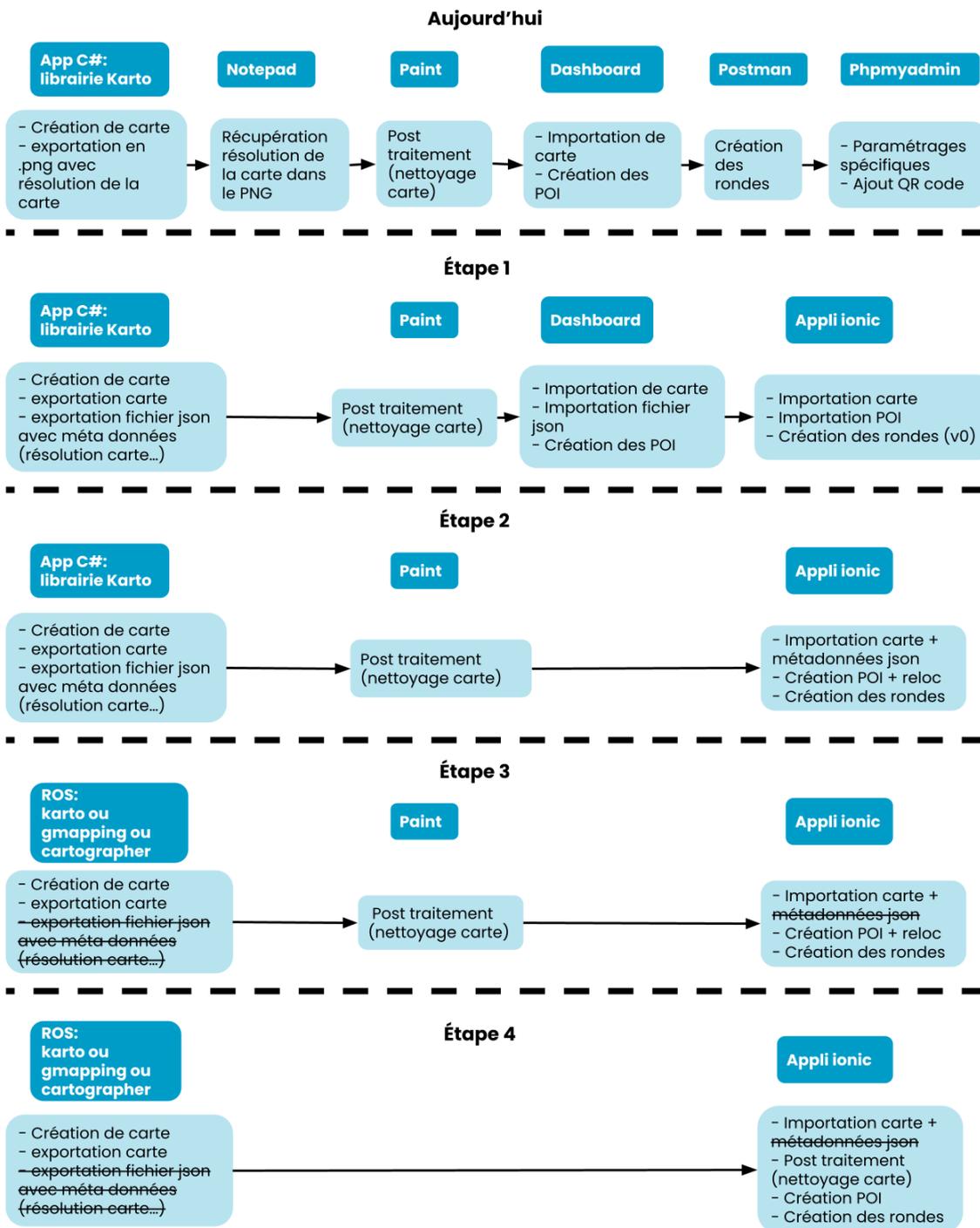


FIGURE 3.6 – Les étapes d’amélioration continue en cours sur le logiciel de cartographie

Ma mission de stage s'inscrit dans le cadre de l'amélioration continue de la version 3 du robot Kompaï, et plus particulièrement dans l'évolution du logiciel de cartographie. À mon arrivée dans l'entreprise, les étapes et les intermédiaires pour que le robot se déplace automatiquement dans son environnement sont nombreux (voir Figure 3.6 étape «Aujourd'hui») :

1. Création de la carte de l'environnement (cartographie) ;
2. Récupération de la résolution de la carte dans le fichier PNG ;
3. Nettoyage de la carte (retirer les impuretés) ;
4. Placement de points d'intérêt (POI<sup>6</sup>) sur la carte pour pouvoir naviguer automatiquement dans l'espace ;
5. Création de rondes, c'est-à-dire une liste de POI qui constitue un circuit fermé à suivre par le robot ;
6. Ajout de QR codes dans l'environnement qui permettent au robot de se repérer ou de se rendre automatiquement à sa borne de recharge.

Le but est donc de réduire au maximum le nombre d'intermédiaires et donc de centraliser un maximum de fonctionnalités. Dans cet objectif, j'interviens dans le passage de l'étape 2 à l'étape 3 (voir Figure 3.6) en remplaçant le logiciel de cartographie actuel qui repose sur une librairie propriétaire Karto.

**Remarque** À mon départ de l'entreprise, le passage de l'étape 3 à l'étape 4 est en cours : l'amélioration est presque terminée.

**D'un point de vue utilisateur**, ce remplacement provient de la volonté de l'entreprise de centraliser tout leur savoir-faire dans la carte électronique principale du robot. Aujourd'hui il est réparti sur deux cartes différentes sous deux OS différents : le logiciel de cartographie sur la tablette Windows 10, et tout le reste sur la carte JETSON TX2 sous Ubuntu 18. Cette centralisation serait d'abord plus pratique pour le développement d'applications, et permettrait d'entamer un premier verrouillage d'accès au savoir-faire de Kompaï.

Effectivement, à l'heure actuelle, le code source est très facile d'accès sur la tablette Windows à n'importe quel utilisateur. Même s'il est protégé par une license, n'importe qui peut le modifier et potentiellement engendrer des problèmes d'exécution ou du plagiat. La carte JETSON protégerait un peu plus l'accès : un utilisateur lambda a trop peu de connaissances robotiques et ne se risquera pas à dévisser le corps du robot pour accéder à la carte. De plus, cela nécessite de brancher quelques périphériques (écran, clavier, souris), et surtout cela nécessite de connaître le mot de passe de session.

**D'un point de vue technique**, le but est de remplacer la librairie propriétaire actuellement utilisée pour cartographier l'espace pour deux raisons :

- Cette librairie ne fonctionne que sous le système d'exploitation Windows, ce qui est incompatible avec le souhait de l'entreprise de centraliser toute la partie logicielle du robot sur la carte électronique principale tournant sous Ubuntu ;
- Il s'agit d'une «boîte noire» puisqu'elle a été achetée il y a quelques années : le code source n'est pas accessible ni modifiable. La librairie n'est donc pas vouée à évoluer ce qui endigue

---

6. Point Of Interest

l'entreprise sur sa volonté de continuer à développer les fonctionnalités et optimiser les performances du robot.

### 3.3.2 Exigences et contraintes

Pour ce changement de logiciel de cartographie, les exigences et contraintes sont multiples :

- Exigences de résultat sur la qualité de la carte obtenue :
  - Des obstacles proprement cernés et définis ;
  - Une géométrie et orientation correcte des objets dans l'espace ;
  - Une épaisseur fine et une netteté des murs ;
  - Une bonne couverture laser, c'est-à-dire peu d'espace cartographié laissé comme "incertain" ou "inconnu" sur la carte finale.
- Contraintes et exigences logicielles :
  - Un système d'exploitation et une distribution définis : GNU/Linux Ubuntu 18.04 qui impose donc ROS 1 Melodic ;
  - Une actualisation de la carte maximum toutes les cinq secondes côté interface utilisateur, ce qui comprend la création, l'enregistrement, l'envoi et l'affichage de la carte ;
  - Un enregistrement de la carte sous format PNG pour pouvoir être correctement traitée et envoyée par la librairie KomNAV via le protocole HTTP à l'interface utilisateur ;
  - L'utilisation de données laser 2D et odométriques uniquement (pas d'IMU disponible sur le robot) ;
  - L'intégration de l'algorithme SLAM choisi avec PURE (accès aux données capteurs) et ROS (publications des données traitées sur les topics dédiés) ;
  - Pourvoir modifier la résolution de la carte.
- Contraintes matérielles :
  - Les ressources RAM/CPU sont limitées (8Go de RAM), il faut donc déterminer le plus grand espace cartographiable possible avec ces données sans entraîner de latence pour l'utilisateur (faire attention aux coûts) ;
  - La ROM de la JETSON est limitée à 32Go dont 28 disponibles et 25 déjà utilisés par Kompaï Roboyics (s'il y a trop d'applications, le risque de ralentissement du processus prioritaire augmente) ;
  - Tout centraliser est pratique mais la performance de calcul est réduite par rapport à une carte dédiée spécifique (logiciel embarqué optimisé au maximum pour ne réaliser qu'une seule tâche). De la même façon, la parallélisation des tâches est plus limitée sur une seule carte que plusieurs.

Toutes ces contraintes et exigences sont donc à prendre en compte lors du choix du logiciel, de son intégration et de ses tests.

## 4 Réalisation

Cette partie décrit l'étendue du travail documentaire et technique réalisé pour mener à bien les objectifs de mission, mais également l'organisation des tâches tout au long de cette mission.

### 4.1 Étude théorique des solutions de SLAM existantes

Cette étude permet d'évaluer l'étendue des possibilités de SLAM, avec leurs avantages et leurs inconvénients, pour pouvoir définir la solution la plus adaptée à mettre en place.

Le SLAM consiste à cartographier un environnement intérieur et se localiser dans cet environnement simultanément. Aujourd'hui, il existe plusieurs types d'algorithmes de SLAM dont certains sont directement utilisables sous ROS. Une étude théorique des solutions existantes disponible en Annexe 7.3 montre que les trois algorithmes les plus prometteurs sous ROS sont Karto, Gmapping et Cartographer.

### 4.2 *Benchmark* des SLAM retenus sous ROS

L'objectif de cette partie est de comparer les trois algorithmes de SLAM retenus en termes de qualité de la carte produite, de rapidité de traitement et de consommation de ressources informatiques. Il faut les comparer entre eux, mais aussi les comparer au logiciel de cartographie déjà utilisé par l'entreprise **Slam2d**. Ce logiciel repose d'ailleurs sur l'algorithme de SLAM Karto mais avec un optimiseur plus performant.

**Remarque** Pour ce *benchmark*, un *workspace* Catkin et un dépôt Bitbucket dédiés sont créés afin de pouvoir facilement déployer et tester le code source sur d'autres robots Kompaï. Une attention particulière est portée à la rédaction du *README* pour faciliter l'installation et l'utilisation du *workspace* sur un robot.

Dans tous les cas, la phase la plus fastidieuse lors de l'implémentation de ces trois algorithmes sur le robot est la paramétrisation [1][2][3]. Chaque paramètre est à étudier selon sa pertinence de modification de la carte en cours de création. Beaucoup de tests ont été réalisés avant d'arriver à une configuration satisfaisante de chacun des trois algorithmes.

Une fois la configuration logicielle faite, pour comparer les trois algorithmes SLAM (Cartographer, GMapping et Karto), cinq pièces de surfaces différentes sont cartographiées :

1. Les locaux de Kompaï Robotics ;
2. Le premier étage au-dessus des locaux de Kompaï Robotics ;

3. Le rez-de-chaussée de l'ESTIA (école d'ingénieur de Bidart) ;
4. Le premier étage de l'EPHAD d'Arcangues ;
5. Le musée Deutsches de Munich (uniquement utilisable pour Cartographer).

**Remarque** Les données du musée Deutsches de Munich ont été recueillies sous forme de *rosbag* avec des données laser issues d'une caméra 3D et une IMU. La conversion des données d'une IMU en données odométriques n'étant pas triviale, ce *rosbag* n'a été testé que avec Cartographer.

Lorsque l'algorithme de SLAM tourne, une fenêtre RViz permet d'observer son avancement (voir Figure 4.1).

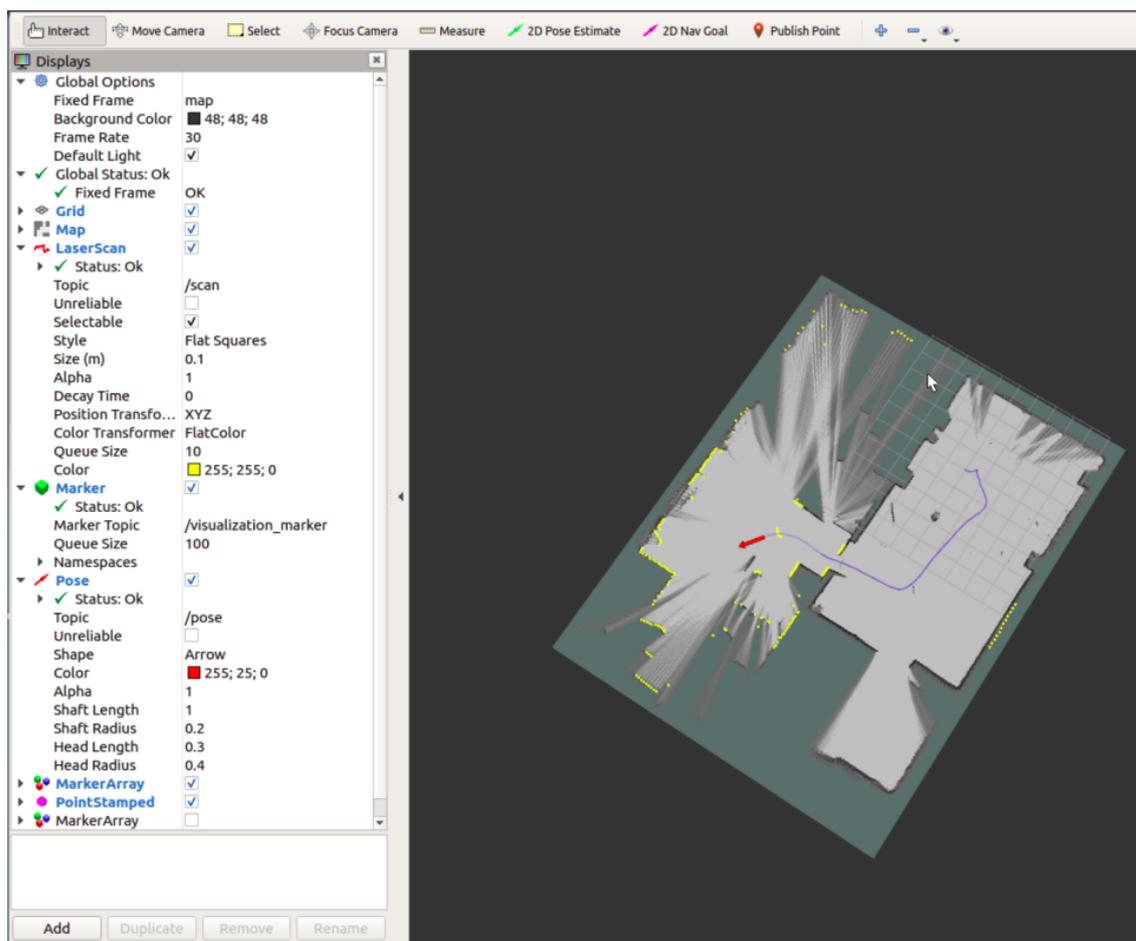


FIGURE 4.1 – Visualisation de la fenêtre RViz lors de la cartographie

### 4.2.1 Temps de traitement

Le temps de traitement entre deux mises à jour de la carte est, pour les quatre algorithmes, tout à fait satisfaisant. Pour les tests, il a été paramétré de 0.5 à 2 secondes pour chacun des quatre algorithmes ; aucune latence n'a été constatée.

Cependant, la sauvegarde de la carte au format PNG est plus longue (0.5 seconde), et l'envoi de la carte par requête HTTP à l'interface utilisateur l'est encore plus (1 seconde). C'est pourquoi le temps entre deux actualisations de la carte par l'algorithme de SLAM est fixé à 2 secondes. C'est avec cette période d'actualisation que toutes les comparaisons ont été faites.

De plus, la résolution de la carte générée a été fixée à 0.05 m par cellule de la grille d'occupation pour avoir des temps de traitement homogènes entre les algorithmes et donc une comparaison valable.

## 4.2.2 Consommation des ressources informatiques

Pour toutes les cartographies réalisées, la consommation de RAM (en Mo) et de CPU (en %) est relevée pour chaque algorithme afin de comparer les coûts.

Le musée allemand est la plus grande surface à ma disposition aujourd'hui. Lors de sa cartographie, **Cartographer** n'a pas planté et la carte a pu être terminée. Par contre, la RAM de la JETSON était presque saturée (7.8/8 Go) sur la fin de la cartographie ce qui entraînait une grosse latence sur la mise à jour de la carte. Avec quelques calculs d'extrapolation des données, la CPU du robot serait plus limitante que la RAM : maximum 12000 m<sup>2</sup> d'espace pourrait être cartographié avec Cartographer, ce qui est largement suffisant pour le besoin de l'entreprise.

**Slam2d** n'est pas comparable avec les autres SLAM car il tourne sur une architecture matérielle et un OS différents de la JETSON (Windows 10 vs Ubuntu 18). En revanche, son évolution relative en fonction de la surface cartographiée est exploitable.

Space	Surface (m <sup>2</sup> )	Karto	Slam2d	Gmapping	Cartographer
Floor 1	211	23	128	87	75
Workshop	234	17	114	18	47
Arcangues floor 1	341	31	148	92	124
ESTIA	1040	33	147	152	115
Deustches Museum	11640	/	/	/	1258

FIGURE 4.2 – Consommation moyenne de RAM en Mo pendant la cartographie

Space	Surface (m <sup>2</sup> )	Karto	Slam2d	Gmapping	Cartographer
Floor 1	211	8	22	16	15 / 63
Workshop	234	4	15	15	15 / 54
Arcangues floor 1	341	8	36	17	12 / 38
ESTIA	1040	5	20	16	15 / 35
Deustches Museum	11640	/	/	/	35 / 65

FIGURE 4.3 – Consommation moyenne de CPU en % pendant la cartographie

L'évolution des valeurs de CPU n'est pas linéairement proportionnelle à la surface cartographiée mais plutôt à la complexité de l'espace à cartographier (voir Figure 4.3). Plus il y aura des

fermetures de boucle à réaliser, plus de ressources CPU seront utilisées pour les réaliser.

Concernant la RAM, l'évolution est globalement proportionnelle à la surface cartographiée quel que soit l'algorithme utilisé (voir Figure 4.2).

Slam2d tourne sous Windows donc la comparaison avec les autres algorithmes testés sous Linux est délicate. Ce qui est sûr c'est que Slam2d, reposant sur l'algorithme Karto, a des consommations relatives similaires à Karto et peine à fermer les boucles, tout comme Karto. Gmapping et Cartographer sont beaucoup plus performants sur cette tâche de fermeture de boucle, et donc consomment beaucoup plus de CPU et de RAM.

La consommation moyenne de RAM et de CPU pour Gmapping et Cartographer est très similaire. Cependant quelques pics de CPU sont notables pour Cartographer (deuxième valeur de la case) lors des fermetures de boucle puisqu'il est beaucoup plus réactif que Gmapping sur la détection des boucles à fermer, et sur la mise à jour de la carte complète après fermeture de boucle.

En effet Karto, tout comme Gmapping, enregistre et travaille avec le grahe complet de la carte générée. Cependant, Karto n'actualise ce graphe que lors de fermetures de boucle, ce qui lui permet de consommer très peu de RAM et de CPU. En revanche, il a plus de mal à fermer les boucles puisque parfois la carte a trop divergé pour que l'optimiseur fasse correspondre correctement les noeuds déjà explorés avec les nouveaux.

Gmapping, lui, actualise beaucoup plus régulièrement le graphe : dès qu'un décalage trop grand (supérieur à un seuil) est constaté entre le graphe actuel et son état précédent, l'optimiseur est lancé pour recalculer le graphe complet. Ce phénomène entraîne bien sûr une consommation de RAM et de CPU plus importante mais les boucles sont bien mieux fermées par la suite dans des environnements intérieurs complexes par rapport à Karto.

Enfin, Cartographer gère des sous-graphes qu'il insère au fur et à mesure de la cartographie dans le graphe complet. Cette stratégie lui permet de ne travailler qu'avec quelques sous-graphes (celui actuel et quelques précédents) pour consommer moins de RAM et de CPU. Cependant, lorsqu'il détecte une fermeture de boucle, il doit alors insérer ces sous-graphes dans le graphe complet ce qui coûte beaucoup mais ponctuellement en CPU.

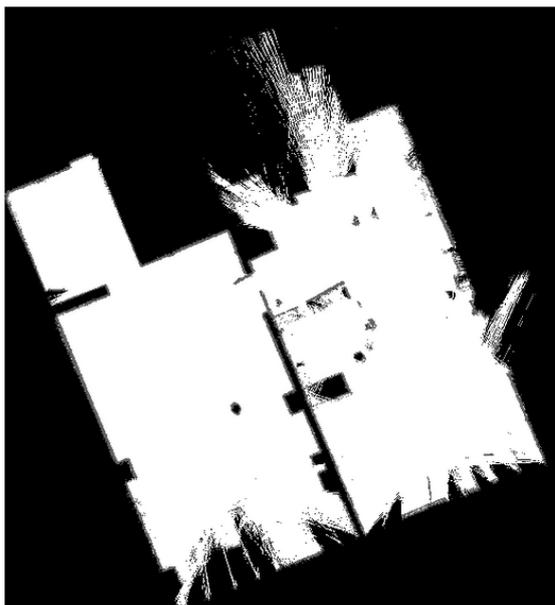
### 4.2.3 Qualité des cartes obtenues

Après une longue paramétrisation pour chaque algorithme, la qualité des cartes obtenues des locaux de l'entreprise avec les quatre SLAM testés sont très similaires (voir Figure 4.4). Elles sont toutes validées car très proches du résultat donné par **Slam2d**.

Concernant le premier étage au-dessus des locaux de l'entreprise, il y a un décalage trop grand sur la carte obtenue avec **Karto** (voir Figure 4.5). Cet algorithme est donc éliminé pour le remplacement de **Slam2d**.

Concernant le rez-de-chaussée de l'ESTIA, il y a un décalage trop grand sur les cartes obtenues avec **Karto** et **Slam2d** (voir Figure 4.6). Les algorithmes **Gmapping** et **Cartographer** donnent donc tous les deux de meilleurs résultats que **Slam2d**. Pour les départager, l'EPHAD d'Arcangues est cartographié.

Après avoir cartographié l'EPHAD d'Arcangues, **Gmapping** et **Cartographer** proposent toujours des résultats très satisfaisants. En revanche, les finitions avec **Cartographer** sont plus propres puisque les faisceaux laser sont lissés (voir Figure 4.7).



Karto



Gmapping



Slam2d



Cartographer



FIGURE 4.4 – Cartes des locaux de Kompai Robotics obtenues avec les différents SLAM

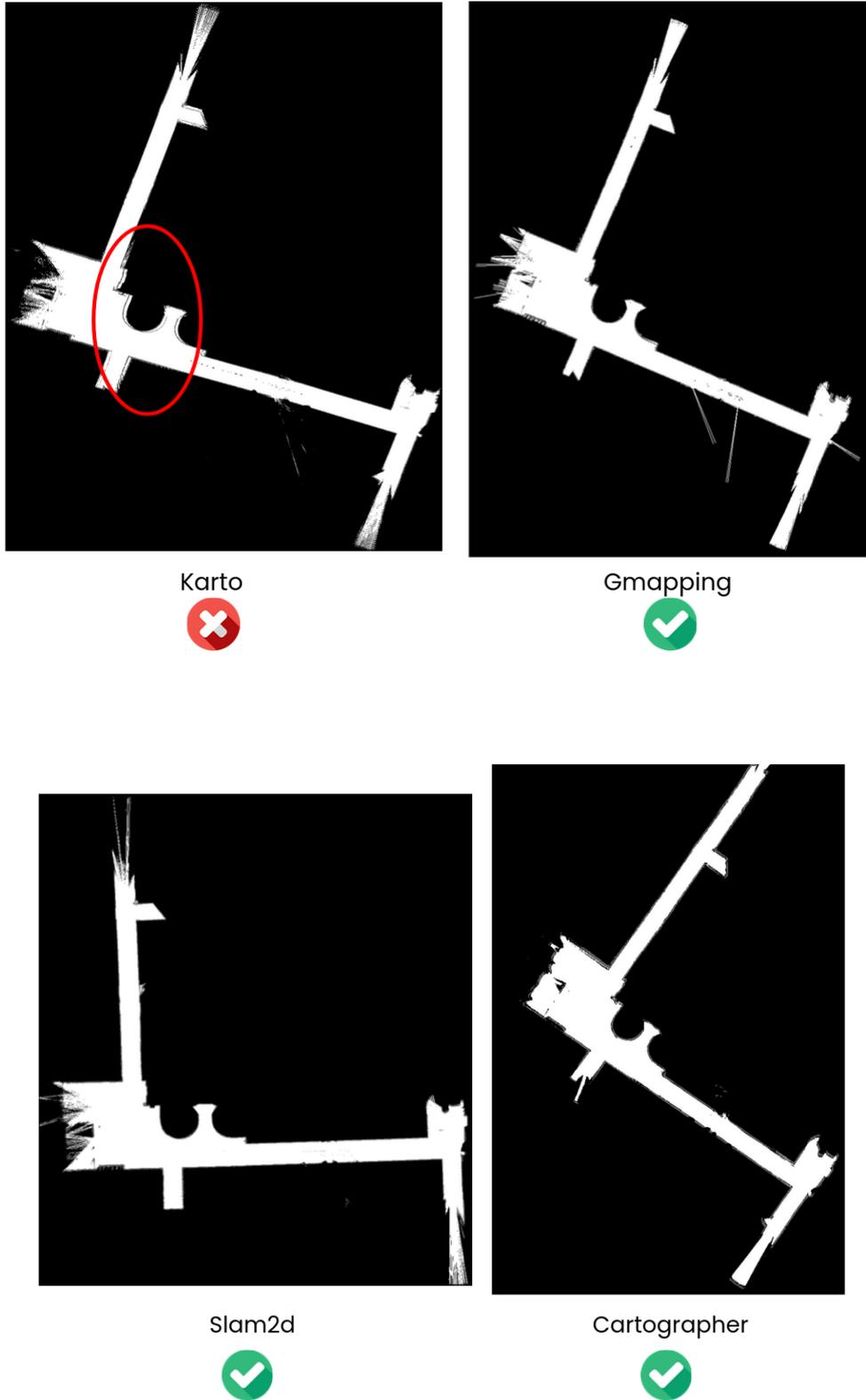


FIGURE 4.5 – Cartes du premier étage obtenues avec les différents SLAM

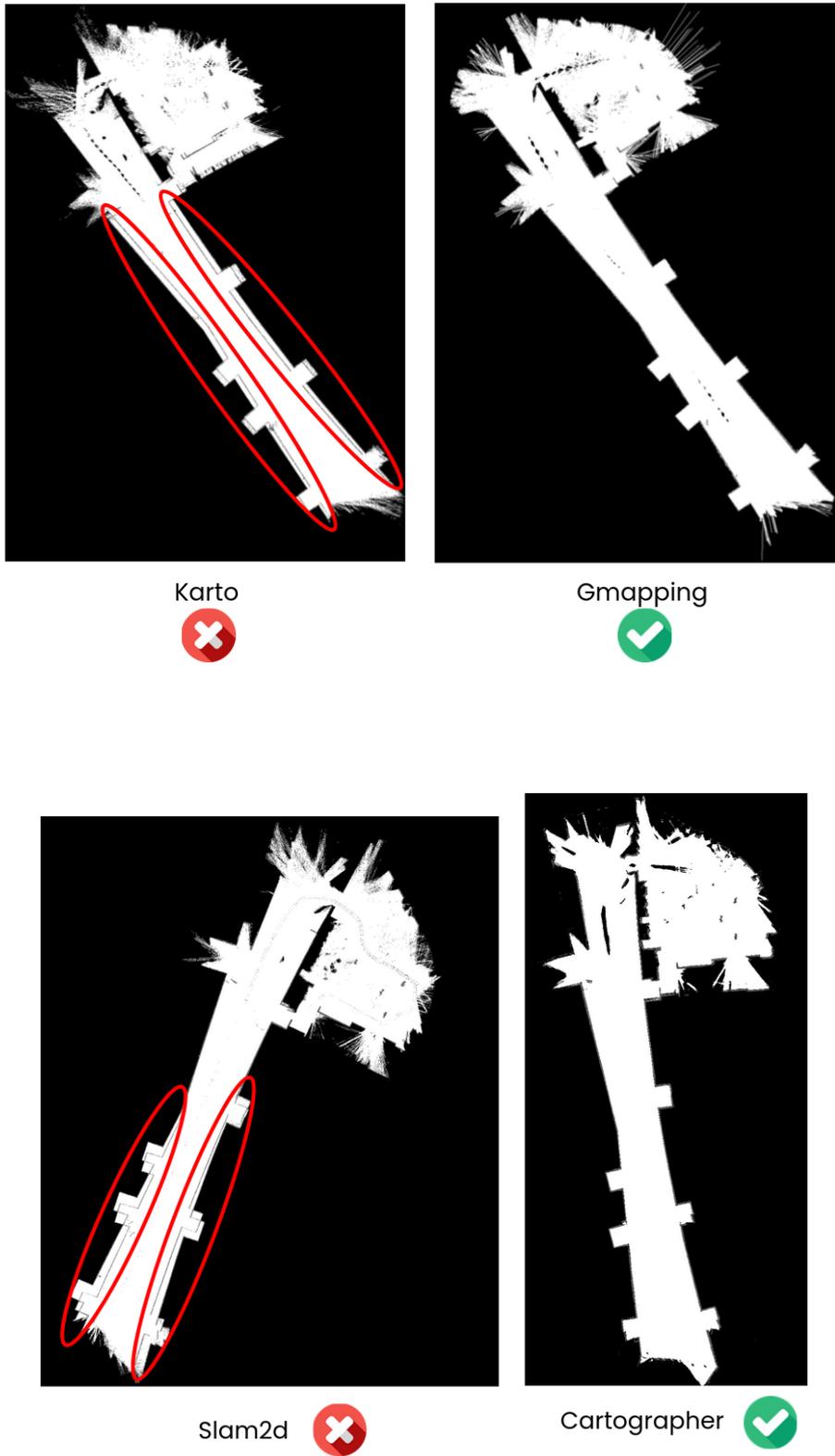
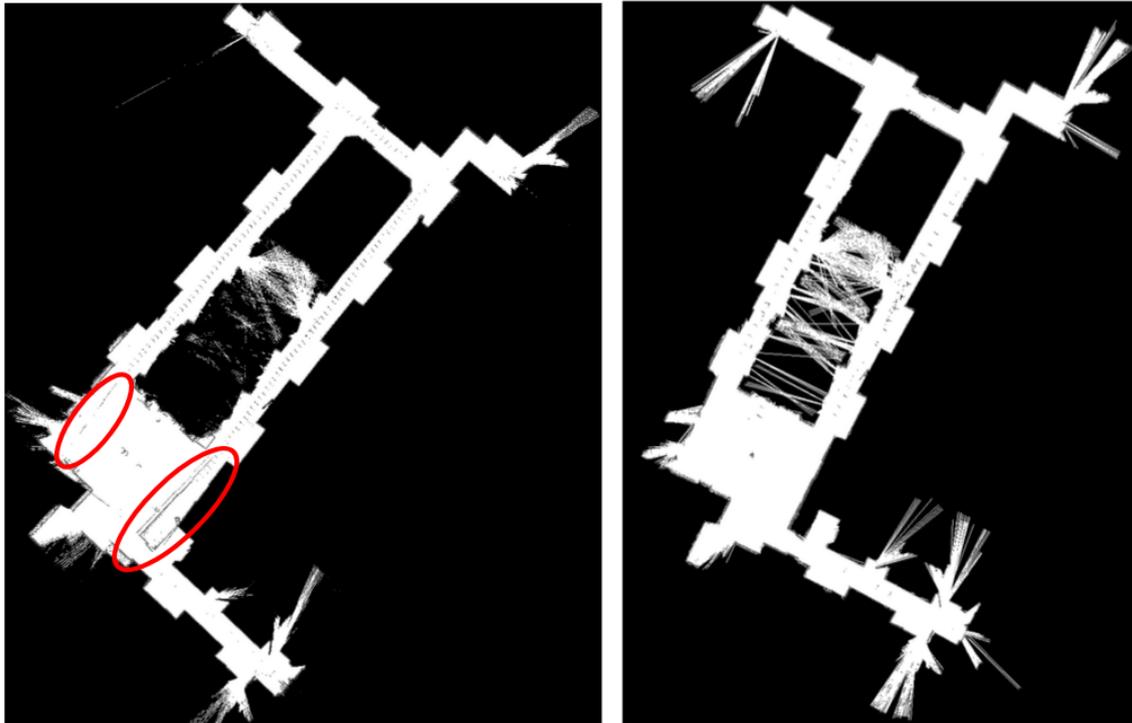
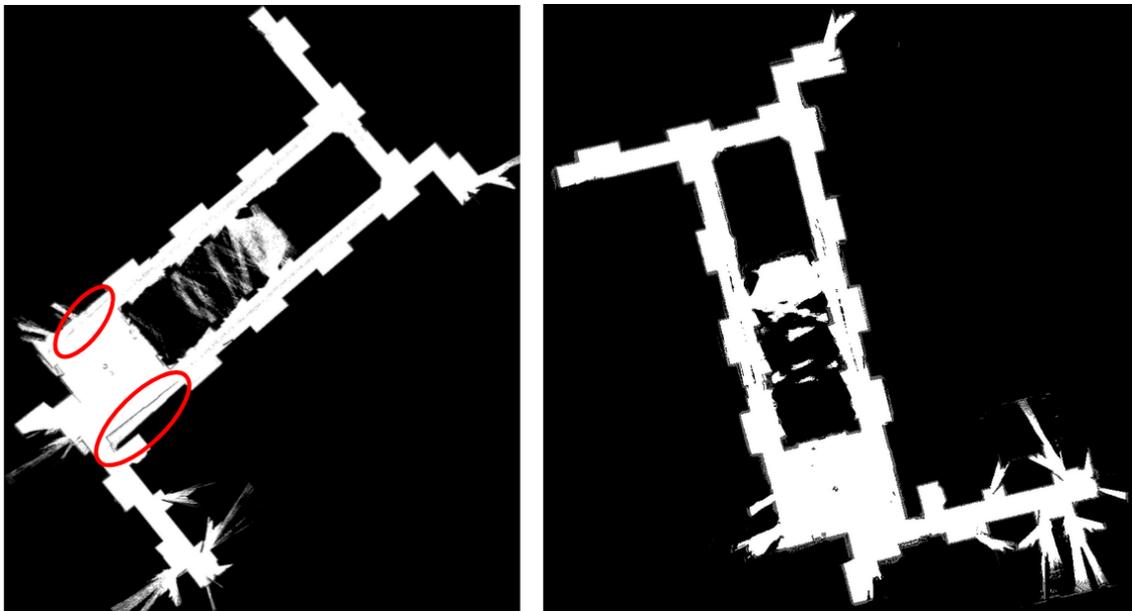


FIGURE 4.6 – Cartes de l'ESTIA obtenues avec les différents SLAM



Karto ❌

Gmapping ✅



Slam2d ❌

Cartographer ✅

FIGURE 4.7 – Cartes de l'EPHAD d'Arcangues obtenues avec les différents SLAM

**Remarque** Quelques traces noires compactes ou régulièrement espacées peuvent être constatées sur le trajet du robot. Elles correspondent à la réflexion des faisceaux du laser sur les surfaces murales particulièrement réfléchissantes (baies vitrées, murs blancs éclairés par la lumière). Une paramétrisation adéquate de l’algorithme SLAM permet de fortement atténuer ce phénomène mais il demeure parfois sous certaines circonstances défavorables. Il faut alors, lors du nettoyage manuel de la carte, retirer ces traces.

**Gmapping** et **Cartographer** sont les deux algorithmes qui sortent du lot puisqu’ils donnent de meilleurs résultats que **Slam2d**. C’est finalement **Cartographer** qui sera choisi pour une propreté de la carte légèrement meilleure (pas ou peu de faisceaux laser traversant les surfaces vitrées), et surtout, pour ses fonctionnalités supplémentaires (possibilité d’intégrer des ancres dans l’environnement pour recalibrer la carte, possibilité de reprendre la cartographie d’une carte déjà existante et sauvegardée sous un certain format).

L’algorithme de SLAM est ensuite intégré dans l’architecture logicielle du robot afin que la cartographie de l’environnement puisse être lancée depuis l’interface utilisateur avec une actualisation toutes les cinq secondes environ (voir Figure 4.8).

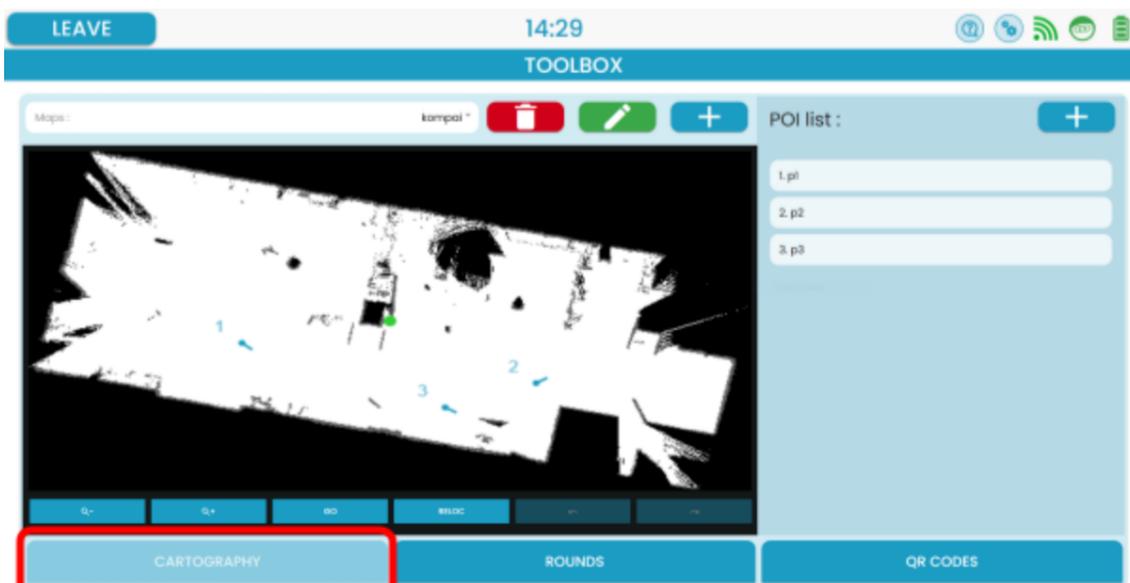


FIGURE 4.8 – Nouvelle IHM de cartographie et de création de rondes

### 4.3 Tâches annexes

De petites missions annexes me sont confiées telles que :

- Faire évoluer l’interface utilisateur de l’ancien logiciel de cartographie **Slam2d** selon le besoin client, et résoudre un problème de latence de mise à jour de la carte à cause de la réécriture d’un fichier PNG dans le logiciel **Slam2d** (voir Annexe 7.4) ;
- Réaliser une petite veille technologique sur un système RTLS<sup>1</sup> de chez *Terabee* (voir Annexe

1. Real Time Localization System

- 7.5);
- Construire des organigrammes de fonctionnement logiciel et matériel du robot Kompaï actuel (voir Annexe 7.6);
  - Intégrer le nouveau logiciel de cartographie et la détection d'obstacles déjà sous ROS avec PURE (voir Annexe 7.7).

## 5 Gestion de projet

Cette partie détaille mon organisation prévisionnelle et effective au cours de ma mission pour s'adapter aux difficultés rencontrées.

### 5.1 Organisation

Tout au long du stage, des réunions hebdomadaires avec mon maître de stage étaient réalisées, et les comptes-rendus étaient écrits sur *Basecamp*, le logiciel de suivi de projet de l'entreprise. C'est également sur ce logiciel que les discussions sur un sujet technique ou la liste des tâches à réaliser sur un projet sont renseignées.

Le planning prévisionnel pour ma mission de stage est le suivant :

Tâche \ Date	Avril	Mai	Juin	Juillet	Août
Compréhension du fonctionnement du robot Kompaï	■				
État de l'art des solutions SLAM existantes	■				
Tests des SLAM sous ROS		■	■		
Comparaison des SLAM sous ROS et choix final			■		
Intégration du SLAM choisi dans le robot Kompaï				■	
Exploration de l'utilisation d'ancres avec le SLAM				■	
Exploration de la reprise de carte avec le SLAM					■
Rédaction de la documentation nécessaire					■

FIGURE 5.1 – Planning prévisionnel pour ma mission de stage

Tâche \ Date	Avril	Mai	Juin	Juillet	Août
Compréhension du fonctionnement du robot Kompaï	■				
Résolution du bug logiciel cartographie Slam2d	■	■			
Diagrammes fonctionnels robot Kompaï			■		
État de l'art des solutions SLAM existantes		■			
Évolution IHM Slam2d			■		
Tests des SLAM sous ROS		■	■	■	
Comparaison des SLAM sous ROS et choix final				■	
Intégration du SLAM choisi dans le robot Kompaï					■
Exploration de l'utilisation d'ancres avec le SLAM					■
Exploration de la reprise de carte avec le SLAM					■
Rédaction de la documentation nécessaire					■

FIGURE 5.2 – Planning effectif pour ma mission de stage

Le planning effectif diffère légèrement du premier (voir Figure 5.2). Effectivement, je n'ai pas

eu le temps d'explorer les ancres sous Cartographer et dont le réajustement de la carte générée en fonction des QR codes flashés dans l'environnement pas le robot. De même, je n'ai pas pu me pencher sur le problème de reprise d'une cartographie à partir d'une carte existante. Cependant, de petites tâches annexes sont venues se greffées à ma mission (en jaune sur la Figure 5.2), ce qui explique ce décalage temporel dans la réalisation de mes tâches.

De plus, je tenais un carnet de bord quotidien sur toutes les actions menées dans la journée, ce qui m'a été très utile pour pouvoir dater des actions ou en rendre compte lors de mes réunions hebdomadaires. J'ai aussi utilisé un tableau *Trello* pour organiser et prioriser mes tâches à réaliser. Cet outil agile - que j'utilise depuis quelques années maintenant - m'est devenu indispensable pour avoir d'un coup d'oeil l'état d'avancement de mes missions, les délais dont je dispose pour les réaliser, les recherches faites et les tests réalisés avec leurs résultats. Cet outil complet et flexible m'a permis de toujours savoir où j'en étais globalement dans ma mission de stage.

## 5.2 Problèmes rencontrés

Les difficultés que j'ai pu rencontrer ont finalement été quasiment que d'ordre technique : j'ai régulièrement fait appel à mes collègues pour comprendre l'architecture logicielle et matérielle du robot, pour obtenir de l'aide sur des erreurs de compilation ROS ou PURE, pour avoir leur avis ou aval sur l'implémentation logicielle que je prévoyais.

J'ai également été en totale autonomie sur les tâches que je devais réaliser, ce qui m'a posé quelques problèmes de prise de décision, de motivation et de prise de recul. En effet, les choix d'implémentation n'étaient dictés que par moi ; j'étais parfois trop indécise et donc trop lente sur ma prise de décision. J'ai également parfois eu du mal à me motiver sur des tâches complexes ou fastidieuses puisque j'étais seule à travailler sur le SLAM. Enfin, il m'a manqué du recul par rapport à mon code lorsque je déboguais ; j'avais souvent une vision étriquée du problème de laquelle j'avais du mal à me détacher. C'est un biais cognitif courant nommé *tunnel vision*. Dans ce dernier cas, la stratégie la plus simple et la plus efficace que j'ai mise en place était de demander l'avis de mes collègues Antoine TRIAUX et Marc TRAONMILIN du pôle R&D.

## 6 Conclusion

Un nouveau logiciel de cartographie de l'environnement a donc été intégré avec succès sur le robot Kompaï, ce qui permet à la fois de répondre aux objectifs de :

- Centralisation et protection du savoir-faire Kompaï Robotics ;
- Simplification du processus de déplacement automatique du robot Kompaï dans un nouvel environnement pour l'utilisateur final ;
- Obtention de cartes de meilleure qualité (fidélité à la réalité et précision) tout en limitant la consommation de ressources informatiques.

Par ailleurs, ayant déjà un diplôme d'ingénieur en Cognitique et ayant déjà réalisé trois stages ingénieur de durées variables en entreprise, cette nouvelle expérience était l'occasion d'élargir mes horizons professionnels et techniques. En effet, ce stage a été enrichissant sur plusieurs aspects :

- Il m'a permis de mettre en pratique des connaissances théoriques vues en spécialité Robotique Mobile à l'ENSTA Bretagne telles que l'implémentation logicielle d'un algorithme SLAM dans un robot mobile dont la configuration matérielle était existante. L'intégration de Cartographer dans l'architecture logicielle du robot Kompaï était challengeante mais a été facilitée par le développement de briques logicielles communiquant entre PURE et ROS.
- Il m'a permis de découvrir le fonctionnement d'une start-up, moi qui avais réalisé mes trois stage en grand groupe (DAHER, Thales) ou PME (KUKA Systems Aerospace). Je me suis ainsi rendue compte que je préférais l'ambiance de travail d'une start-up ou d'une PME que d'un grand groupe.
- Le stage était appliqué à la robotique mobile, là où mon précédent stage de fin d'études concernait la robotique industrielle manipulatrice (bras robotiques assembleurs de pièces aéronautiques sur les lignes de production d'avion). Ce stage a confirmé que les défis et enjeux liés à la robotique mobile m'intéressent davantage que ceux de la robotique manipulatrice.
- Enfin, j'ai pu me rapprocher du domaine de la robotique médicale auquel j'aspire depuis un an et demi maintenant puisqu'il s'agissait de robotique sociale destinée aux patients et aux personnel soignant. Le robot aide à la marche des personnes à mobilité réduite, mais réalise également des rondes de surveillance pour alerter le personnel en cas de chute d'une personne.

Ce stage a donc été une expérience humaine et professionnelle riche dans un contexte technologique innovant.

# Bibliographie

- [1] Yassin Abdelrasoul, Abu Bakar Sayuti HM Saman, and Patrick Sebastian. A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 1–6, Ipoh, September 2016. IEEE.
- [2] Dongxiao Han, Yuwen Li, Tao Song, and Zhenyang Liu. Multi-Objective Optimization of Loop Closure Detection Parameters for Indoor 2D Simultaneous Localization and Mapping. *Sensors*, 20(7) :1906, March 2020.
- [3] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2D LIDAR SLAM. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, Stockholm, Sweden, May 2016. IEEE.
- [4] Camden Mark Koehr. EVALUATION OF ROS SLAM GMAPPING FOR EXTRATERRESTRIAL ROBOTIC MINING. page 154.
- [5] Maxim Kuzmin. Review. Classification and Comparison of the Existing SLAM Methods for Groups of Robots. In *2018 22nd Conference of Open Innovations Association (FRUCT)*, pages 115–120, Jyvaskyla, May 2018. IEEE.
- [6] Eric Demeester Peter Aerts. Benchmarking of 2D-SLAM Algorithms a Validation for the TETRA project Ad Usum Navigantium. July 2017.
- [7] Monserrat Rojas-Fernandez, Dante Mujica-Vargas, Manuel Matuz-Cruz, and Diana Lopez-Borreguero. Performance comparison of 2D SLAM techniques available in ROS using a differential drive robot. In *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 50–58, Cholula, February 2018. IEEE.
- [8] P Sankalprajan, ThiriloChan Sharma, Hamsa Datta Perur, and Prithvi Sekhar Pagala. Comparative analysis of ROS based 2D and 3D SLAM algorithms for Autonomous Ground Vehicles. In *2020 International Conference for Emerging Technology (INCET)*, pages 1–6, Belgaum, India, June 2020. IEEE.
- [9] Joao Machado Santos, David Portugal, and Rui P. Rocha. An evaluation of 2D SLAM techniques available in Robot Operating System. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6, Linkoping, Sweden, October 2013. IEEE.
- [10] Xuexi Zhang, Jiajun Lai, Dongliang Xu, Huaijun Li, and Minyue Fu. 2D Lidar-Based SLAM and Path Planning for Indoor Rescue Using Mobile Robots. *Journal of Advanced Transportation*, 2020 :1–14, November 2020.

## 7 Annexes

### 7.1 Robosoft Consulting

#### 7.1.1 La Poste

Le but était réaliser un chariot suiveur de personne pour la distribution de colis (voir Figure 7.1). Ce projet a été effectué dans le cadre d'une réponse à un appel d'offre fait par La Poste afin de produire en 400 exemplaires le chariot suiveur correspondant à leurs attentes. La demande était « un prototype de chariot suiveur propre à la distribution des colis et des lettres par les facteurs piétons dans le centre des villes denses ». L'intérêt pour La Poste était de délester ses facteurs du poids des colis pendant leur tournée dans les zones piétonnes.



FIGURE 7.1 – Chariot suiveur La POSTE (modélisation à gauche et prototype rendu à droite)

Le robot qui a été conçu pour ce projet est capable de suivre une personne précise grâce à un système de géolocalisation d'un tag basé sur du signal UWB<sup>1</sup> de chez IIDRE. L'évitement d'obstacle était possible grâce à une caméra de profondeur.

#### 7.1.2 APRR

Le groupe autoroutier APRR a fait appel à Kompaï Robotics pour obtenir un prototype (voir Figure 7.2) de FLR-SA<sup>2</sup> démontrant la faisabilité d'une flèche de rabattement asservie à la position d'un véhicule pilote. Il s'agit donc d'une remorque autonome déportée par rapport au

---

1. Ultra Wide Band

2. Flèche Lumineuse de Rabattement – Suiveuse Autonome

véhicule à suivre.



FIGURE 7.2 – Robot suiveur pour signalisation de travaux sur autoroute

L'enjeux de ce projet est important puisqu'il s'agit de protéger le personnel du risque routier lors de travaux sur autoroute. En effet, les ouvriers prennent des risques pendant le déplacement des panneaux de signalisation. Ce risque pourrait être fortement diminué si les panneaux pouvaient se déplacer seuls en fonction de l'avancement du chantier.

### 7.1.3 Bio-UV

L'objectif de ce projet était de produire en masse (200 exemplaires) d'un robot mobile de désinfection de locaux (voir Figure 7.3). En effet, dans le contexte de la pandémie, une demande accrue de robots désinfecteurs est apparu. Cette désinfection s'opère soit par un nébulisateur qui vaporise les surfaces avec un agent chimique désinfectant, soit par des lampes à rayonnement UV<sup>3</sup> qui éradique tous les agents bactériens présents sur les surfaces rencontrées. Le projet s'inscrit dans le second cas qui a l'avantage d'avoir une plus grande portée de désinfection, mais qui nécessite l'absence de personne dans la pièce à cause de la dangerosité des rayons. L'utilisation d'un robot est alors toute indiquée.

---

3. Ultra Violet



FIGURE 7.3 – Robot Kompai portant le module de désinfection Bio-UV

Côté *Robosoft Consulting*, l'entreprise travaille aussi sur un exosquelette robotisé pour l'armée, un véhicule de reconnaissance pour l'armée et une plateforme autonome truffée de capteurs destiné aux étudiants l'université de Mulhouse.

## 7.2 Contexte économique et social

### 7.2.1 Étude de marché

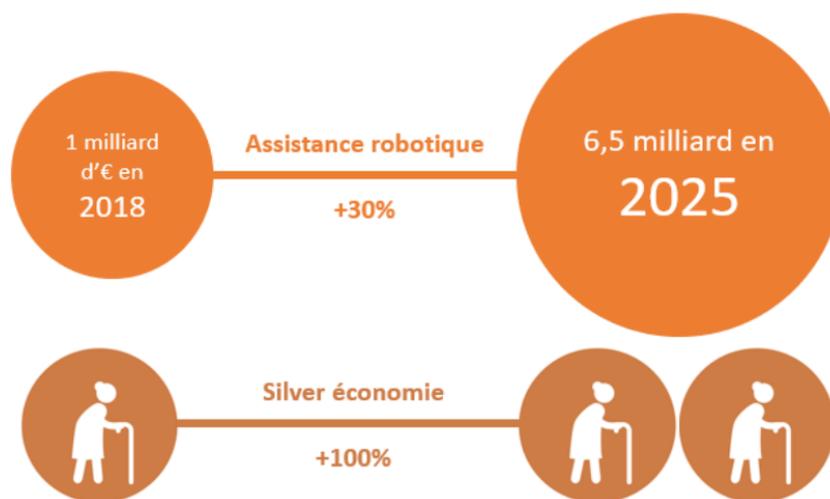


FIGURE 7.4 – Évolution du marché de la robotique d'assistance entre 2018 et 2025

Les chiffres de l'OCDE<sup>4</sup> montrent qu'il manquera 10 millions d'aidants professionnels en Europe en 2050. Les personnes de plus de 65 ans représenteront un tiers de la population Française. Pour répondre à ce problème, la robotique, qui arrive aujourd'hui à maturité, présente un fort potentiel. La Japan Robot Association annonce que la robotique de service représentera en 2025 80% du marché global de la robotique, avec notamment en tête la robotique d'assistance aux personnes (marché estimé à 6,5 milliards d'euros en 2025, voir Figure 7.4).

Depuis sa création en 2016, Kompaï Robotics s'appuie sur le constat de l'OCDE. Pour le moment à un prix de 20 000 €, le robot Kompaï est avant tout destiné aux établissements désireux de se faire accompagner dans le démarrage de leur transition robotique, mais aussi aux intégrateurs de solutions, grâce à des possibilités de personnalisation, tant sur le plan matériel que logiciel. Il répond en priorité aux besoins des établissements dans les tâches répétitives, en aidant les personnels soignants et non en les remplaçant. Il est d'abord conçu pour des actions quotidiennes de routine, comme les tournées d'animation, de surveillance, la détection de chute ou la distraction des résidents en EHPAD.

Grâce à ces fonctions, il présente un retour sur investissement de l'ordre de quelques mois. Son principal atout est sa conception modulaire et multifonctionnel. Il peut être reconfiguré rapidement en cas de situation de crise, en lui ajoutant des éléments complémentaires (lampes UV, pulvérisateurs, capteurs, plateaux repas ...). Plusieurs scénarios sont alors possibles en fonction des besoins (surveillance, aide à la marche, détection de chute, détection du port du masque, prise de température, divertissement, télé-présence, levée de doute, téléconsultation, lien social, désinfection UV, transport de linge, transport de repas ...).

4. Organisation de coopération et de développement économiques

S'il apparaît évident que l'usage de robots va se développer, il n'est pas concevable pour les établissements de s'équiper de plusieurs types de robots différents. Se doter de robots multifonctions, tel que le Kompaï, pouvant être reconfigurés rapidement pour faire face à des imprévus comme le COVID-19, est la stratégie à adopter.

Si la version actuelle du robot est destinée aux EHPAD, la start-up souhaite proposer dans les prochaines années, un robot moins cher pour le maintien à domicile. En effet, en France, on dénombre environ 12 millions de personnes de plus de 65 ans en 2015 et 96 millions au niveau européen. 90% des seniors ont internet et 90% souhaitent vieillir chez eux. Le maintien à domicile est donc une alternative indispensable aux hébergements de type EHPAD ou maison de retraite, aussi bien d'un point de vue humain que d'un point de vue financier.

Il s'agit d'un enjeu majeur pour nos gouvernements qui ne peuvent pas financer l'afflux massif de personnes âgées dans des structures spécialisées. Cependant l'isolement social, qui concerne aujourd'hui plus de 8,5 millions de personnes, engendre des phénomènes de dépression, de perte d'autonomie et de repli sur soi. L'objectif de la prochaine version du Kompaï sera de favoriser le lien social, afin d'améliorer le bien-être et le bien vieillir. Car aujourd'hui, il n'est plus seulement question de rester en bonne santé ou bien de vivre le plus longtemps possible, mais de « bien vieillir » en gardant, malgré l'apparition de pathologies liées à l'âge, son lien social et un maximum d'autonomie.

Le robot Kompaï semble avoir un bel avenir, mais doit faire face à quelques concurrents. Le robot Nao, de la société SoftBank Robotics, est présent à l'internationale dans plusieurs établissements de santé. Dans les EHPADs, il est utilisé pour divertir les personnes âgées en racontant des histoires ou en animant des séances de sport (voir Figure 7.5). Intéressant pour son prix de 5000€, ce robot ne répond que partiellement aux besoins de ces établissements. Contrairement au Kompaï, sa petite taille ne lui permet pas d'apporter un soutien physique aux personnes âgées et est incapable de se déplacer de façon autonome dans les établissements.



FIGURE 7.5 – Robot Nao qui anime une séance de sport dans un EHPAD

En France, le principal concurrent de Kompai est le robot Cutii, de la société CareClever, qui commence à s'installer dans une vingtaine d'EHPAD de la région nord (voir Figure 7.6). A un prix de 6 000 €, Cutii est abordable aussi bien pour les particuliers que pour les établissements.



FIGURE 7.6 – Robot Cutii de la société CareClever

Il s'agit d'un robot compagnon qui permet de rompre l'isolement. Grâce à sa tablette, il est possible de communiquer par appel vidéo avec ses proches, faire des activités de chez soi en direct avec des animateurs et enfin consulter un médecin en quelques minutes. Pour le moment, il ne peut se déplacer de façon autonome ni aider les résidents à marcher. Néanmoins, la société Kompai Robotics a de quoi s'inquiéter car le robot Cutii commence à proposer des fonctionnalités intéressantes comme la détection de chute et le contrôle à distance du robot pour la levée de doute.

### 7.2.2 Enjeux de développement durable

Le robot d'assistance Kompai s'inscrit complètement dans le cadre du pilier social du développement durable. En effet, dans un contexte de vieillissement de la population ce projet a pour ambition d'assister les aidants sans les remplacer. Le bénéfice est double car en allégeant le travail des aidants on diminue la pression subie par ses derniers et on assure de meilleurs soins pour les personnes fragilisées.

De plus, à terme, ce robot pourra également servir à des personnes âgées vivant seules chez elles, leur permettant ainsi de prolonger leur autonomie et de vivre dans leur domicile le plus longtemps possible. Grâce au mode « Assistance à la mobilité » les personnes pourront continuer de marcher seul tout en étant guidées vers la pièce de leur choix. Les robots Kompai permettent aussi de lutter contre l'isolement des personnes âgées et de préserver les liens sociaux grâce aux appels vidéo. Les enjeux sociaux ayant attrait à la sécurité sont également pris en compte. En effet, le robot Kompai est équipé de caméras situées en bas du robot, ce qui permet de détecter les chutes. Une personne référente ou les secours peuvent donc être appelés rapidement. Il est alors possible de prendre le contrôle du robot à distance pour rassurer la personne âgée en attendant

l'arrivée du SAMU.

Mise à part le recyclage de certains composants des robots, on peut difficilement parler d'entreprise verte. En effet, aujourd'hui, la priorité est de faire bon marché et fonctionnel. Le matériel choisi vient la plupart du temps de Chine et n'est pas forcément celui qui consomme le moins d'énergie. Si l'entreprise vient à se développer, elle devra se poser la question d'un développement qui prend en compte une dimension écologique.

### 7.2.3 Application de robotique sociale et médicale

La distanciation sociale imposée par la COVID-19 a créé un besoin de générer un nouveau lien social. La robotique de service va en ce sens : il s'agit de services à la personne qui donnent une interaction avec le personnel médical, avec les autres patients en EHPAD, avec le robot et avec les proches à travers des appels vidéo. Cependant, avec la pandémie, les coupures budgétaires du gouvernement pour les hôpitaux et les services de santé se sont révélées très inquiétantes. Le personnel médical est à bout, se met en grève et demande à pouvoir travailler dans de bonnes conditions en ayant notamment plus de matériel à leur disposition.

En EHPAD, ce sentiment est le même. La première vague a été très difficile pour le personnel médical qui a parfois dû entreprendre des euthanasies douces et cachées pour faire partir certains patients susceptibles d'avoir contracté la COVID-19. Eux aussi réclament plus de personnel et de matériel et se sentent négligés quand les directeurs préfèrent investir dans un robot d'assistance aux personnes à mobilité réduite. Eux y voient un moyen simple de gagner de l'argent : le robot est capable d'effectuer une thérapie douce par des tournées lumineuses qui consistent à passer de la musique et à projeter des animations relaxantes et mouvantes au plafond. Ceci diminue l'anxiété et le stress de certains patients et leur permet de maintenir une certaine activité cérébrale. La prise de médicaments palliant à certains troubles est donc réduite, et de ce fait le budget pour ces médicaments également. Les directeurs sont gagnants.

En revanche, le personnel médical ne le voit pas de cet œil, il voit une mesure qui, une fois de plus, ne va pas dans le sens de leurs réclamations, ce qui est vrai. Les directeurs investissent une et une seule fois pour obtenir un tel robot alors que l'embauche de nouveaux employés leur coûte beaucoup plus cher puisqu'elle se traduit sur le long terme. Le personnel médical rejette alors d'autant plus l'utilisation de ces robots et se montre réticent, voire récalcitrant, face à leur intégration dans l'établissement.

À ceci s'ajoute la réticence initiale des patients, qui sont âgés et qui ont donc naturellement perdu une partie de leur réactivité cognitive et de leur capacité d'apprentissage. Certains n'osent pas utiliser le robot, certains ne savent pas s'ils ont droit de l'utiliser, d'autres ne savent pas comment l'utiliser, d'autres ont peur de ces nouvelles technologies. Certains croient même que le robot serait capable de voler des plateaux repas au réfectoire...

## 7.3 Étude théorique des solutions de SLAM existantes

### 7.3.1 La méthode SLAM

#### 7.3.1.1 Présentation

Le SLAM complet est une solution au problème de la navigation autonome dans un environnement inconnu[4]. Le SLAM consiste à utiliser un robot mobile pour cartographier une zone non répertoriée tout en utilisant cette carte à des fins de localisation dans un premier temps, et de navigation dans un second.

Le SLAM est généralement utilisé dans un environnement clos intérieur mais il a été étendu à des environnements plus complexes tels que des terrains non planaires, des environnements extérieurs (aérien ou maritime), ou des environnements dynamiques. La première solution proposée à ce problème de navigation autonome, qui est aussi la plus répandue, utilise un filtre de Kalman étendu (EKF<sup>5</sup>).

#### 7.3.1.2 Principe

Un système de localisation utilise des balises géométriques comme points de repère et fait correspondre leur observation par un laser du robot à leur emplacement connu dans la carte. Dans ce cas, le robot connaît déjà l'emplacement des points de repère dans l'environnement dans lequel il évolue. Cette connaissance permet de faire correspondre les données des capteurs des points de repère observés avec les positions connues. Toute divergence entre l'endroit où les capteurs indiquent que les points de repère se trouvent et la position réelle des points de repère, indique que la position du robot doit être mise à jour. La partie localisation de la méthode SLAM basée sur EKF nécessite alors des données odométriques fournies par le robot.

Un système de cartographie également basé sur le laser pour un robot mobile est associé au processus de localisation. En général, le système prend des lectures scanner pour créer la carte. Il utilise des relevés provenant de positions et d'angles différents et les intègre dans une grille d'occupation unique de l'environnement. La combinaison des solutions de localisation et de construction de carte permet de localiser le robot et de cartographier l'environnement simultanément.

Pour que le SLAM fonctionne, des données environnementales (**extéroceptives**) et celles du robot (**proprioceptives**) doivent être disponibles. Chaque type de données peut être généré de multiples façons. Les solutions les plus courantes utilisent des capteurs extéroceptifs tels que le LiDAR<sup>6</sup>, le sonar ou même des caméras pour générer des données sur l'environnement. Les données d'odométrie proprioceptives, elles, sont souvent générées par l'utilisation d'encodeurs de roue ou la fusion de données supplémentaires provenant d'une IMU<sup>7</sup>.

---

5. Extended Kalman Filter

6. Light Detection and Ranging

7. Inertial Measurement Unit

Les données ainsi générées sont ensuite utilisées pour construire une carte et se localiser sur cette carte. Une fois les points de repère traités, l'algorithme SLAM tente d'associer ces nouvelles données à tous les points de repère précédemment observés. Ces points de repère peuvent être des obstacles, des angles de mur, etc. Pendant que le robot continue de parcourir et de scanner l'environnement, le processus SLAM utilise en parallèle les données d'odométrie pour estimer la position actuelle du robot sur la carte.

Cette estimation est comparée aux relevés environnementaux effectués par les capteurs et corrigée si elle est inexacte. Le processus est itéré jusqu'à ce qu'une carte complète de l'environnement soit créée. Cet algorithme SLAM générique peut alors être divisé en cinq parties principales[4] :

1. **L'extraction des points de repère** - Identification de points de repère avec les capteurs extéroceptifs pour créer la carte ;
2. **L'association des données** - Recherche des correspondances entre les différents scan laser à travers la reconnaissance des points de repère ;
3. **L'estimation de l'état du robot** - EKF en phase de prédiction ;
4. **La mise à jour de l'état du robot** - EKF en phase de correction ;
5. **La mise à jour des points de repère** - Insertion de la sous-carte dans la carte complète.

Une description de ce processus est présentée visuellement dans la Figure 7.7 ci-dessous.

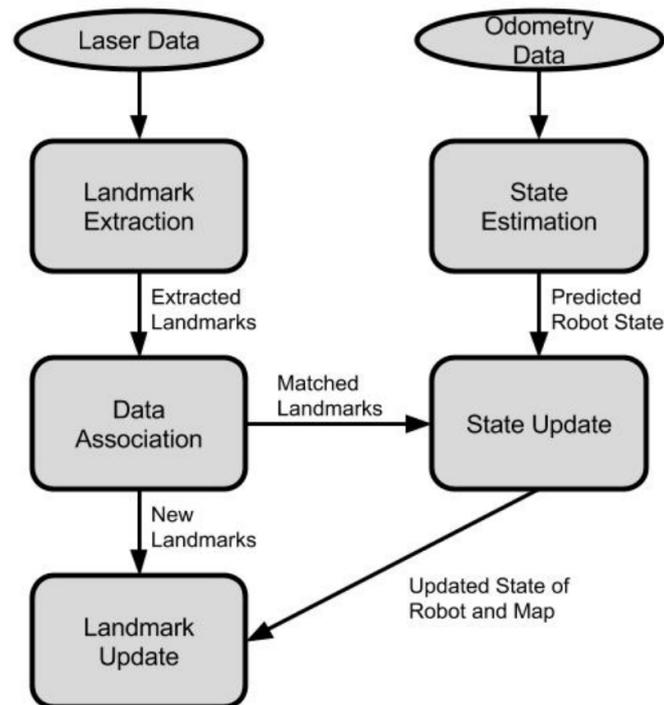


FIGURE 7.7 – Organigramme du processus SLAM

**Remarque** L'étape de mise à jour de l'état dans la Figure 7.7 est synonyme de mise à jour du filtre.

Les différentes approches SLAM suivent généralement la même structure que ci-dessus, sauf que différents filtres sont utilisés dans l'étape de mise à jour de l'état. La figure montre que les données laser et odométriques sont utilisées pour générer des emplacements de repères et un état prédit. Le processus tente ensuite d'associer les nouvelles données laser aux points de repère précédemment observés. Cette étape permet à l'algorithme d'ajuster les erreurs éventuelles dans l'estimation de la position générée par les données odométriques. Les données sont ensuite passées à travers un filtre pour éliminer les erreurs indésirables dans la localisation. Puis le processus est répété. Les sections suivantes sensibilisent aux autres solutions SLAM développées à partir de cette méthode initiale.

### 7.3.2 Les différents SLAM

Aujourd'hui, les différents algorithmes de SLAM diffèrent sur la méthode adoptée pour la localisation et peuvent être regroupés en trois grandes catégories[8][5] (voir Figure 7.8) :

- Les méthodes par **filtrage bayésien** (approche probabilistique) ;
- Les méthodes par **optimisation de graphe** (de la trajectoire du robot) ;
- Les approches basées sur le traitement d'image et l'**intelligence artificielle** (*machine learning* et réseaux de neurones).

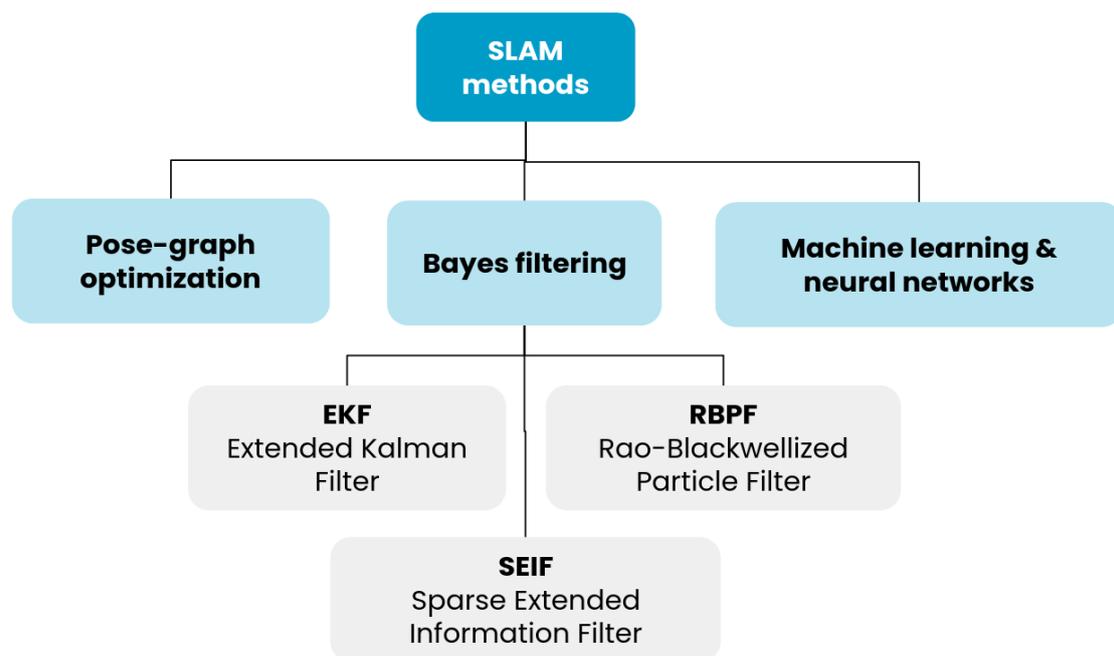


FIGURE 7.8 – Classification des différentes méthodes SLAM

#### 7.3.2.1 Filtrage Bayésien

Les méthodes de filtrage, historiquement, ont été les premiers algorithmes SLAM qui ont été développés à l'époque où les moyens techniques apparaissaient (capteurs imprécis et coûteux).

Ces méthodes sont basées sur l'appareil probabiliste du :

- Le filtre de Kalman étendu (EKF) - la méthode initiale ;
- Le filtre à particules de Rao-Blackwellized (RBPF) ;
- Le filtre d'informations éparses étendu (SEIF)

L'expérience a montré que l'EKF est devenu obsolète et n'est pratiquement plus utilisé en raison des limitations sévères de la taille de la carte et de la faible compatibilité avec les télémètres laser de haute précision.

Parallèlement, le filtre particulaire est largement utilisé et une grande variété de méthodes basées sur ce filtre sont caractérisées par une précision et une robustesse d'estimation élevées.

### 7.3.2.2 Optimisation de graphe

Les méthodes d'optimisation de graphe de position traitent et stockent toutes les données collectées au cours du mouvement d'un robot sous la forme de sommets du graphe. C'est pourquoi la carte ne peut pas toujours être estimée en temps réel. Cependant, ces méthodes sont les plus précises en termes d'estimation de position et, contrairement aux autres classes de méthodes, permettent d'obtenir l'estimation de la totalité du chemin, c'est-à-dire la solution du problème SLAM complet.

### 7.3.2.3 Vision et intelligence artificielle

Les méthodes basées sur l'apprentissage automatique et les réseaux neuronaux sont de plus en plus populaires aujourd'hui mais semblent déjà avoir une grande portée. Leur principal avantage et inconvénient est la nécessité d'un entraînement préliminaire du modèle.

En effet, les données visuelles sont séparées en deux lots :

- Un jeu de données d'entraînement (environ 80% des données) ;
- Un jeu de données de test (les 20 % restants).

La méthode est découpée en deux phases :

1. La **phase d'entraînement** permet de faire un apprentissage du modèle de neurones avec des données visuelles disponibles déjà labellisées à un endroit spécifique de l'environnement. Le réseau de neurones est entraîné à bien associer un motif visuel avec l'endroit correspondant.
2. La **phase de test** qui permet de tester le modèle entraîné sur les données restantes pour contrôler que les estimations de localisation du robot dans son environnement sont correctes. Cette phase valide ou invalide le modèle selon le pourcentage de réussite.

En phase de test, s'il n'y a eu pas de sous-apprentissage ou de sur-apprentissage lors de la phase d'entraînement, l'algorithme permettra de se localiser très rapidement et précisément. En revanche, la phase précédente d'apprentissage est fastidieuse et coûteuse.

### 7.3.3 Comparaison théorique des différents SLAM 2D sous ROS

Il existe des méthodes de résolution du problème SLAM en 3D et en 2D. Ma mission se concentre uniquement sur ceux 2D. De plus, pour cette comparaison, les méthodes SLAM 2D utilisées font partie du filtrage bayésien ou de l'optimisation de graphe.

#### 7.3.3.1 CRSM SLAM

CRSM<sup>8</sup> SLAM utilise la correspondance entre le balayage laser et la carte déjà existante pour poursuivre la construction de la carte 2D[7]. Il effectue un prétraitement du balayage et sélectionne les rayons qui sont critiques pour le processus de correspondance. Il s'agit donc d'un filtre bayésien et, plus particulièrement, d'un filtre particulaire.

#### 7.3.3.2 TinySLAM et CoreSLAM

TinySLAM est un algorithme de SLAM de 200 lignes de code, qui a été créé dans le but d'être simple et facile à comprendre avec une perte minimale de performance. C'est également un filtre bayésien et, plus particulièrement, un filtre particulaire. Il utilise également un balayage laser pour construire la carte et des données odométriques pour se localiser[6].

CoreSLAM est un wrapper ROS pour l'algorithme original TinySLAM[9]. L'algorithme est divisé en deux étapes différentes :

1. Le calcul de la distance aux objets de l'environnement ;
2. La mise à jour de la carte.

Dans la première étape, pour chaque balayage entrant, il calcule la distance sur la base d'un algorithme de filtre particulaire très simple. Le filtre particulaire fait correspondre chaque scan du laser avec la carte et chaque particule du filtre représente une position possible du robot et a un poids associé, qui dépend des itérations précédentes. Après la sélection des meilleures hypothèses, les particules ayant un poids plus faible sont éliminées et de nouvelles particules sont générées.

Dans l'étape de mise à jour, les lignes correspondant aux scans reçus sont dessinées dans la carte. Cependant, au lieu de dessiner un seul point lorsqu'un obstacle est détecté, TinySLAM dessine un ensemble réglable de points entourant l'obstacle.

#### 7.3.3.3 LagoSLAM

LagoSLAM fait partie des algorithmes SLAM par optimisation de graphe. La base des algorithmes SLAM basés sur les graphes est la minimisation d'une fonction de coût non linéaire et non convexe. Plus précisément, à chaque itération, une approximation convexe locale du problème initial est résolue afin de mettre à jour la configuration du graphe. Le processus est répété jusqu'à ce qu'un minimum local de la fonction de coût soit atteint[9].

Cependant, ce processus d'optimisation est fortement dépendant d'une estimation initiale pour converger. Une nouvelle approche appelée LAGO<sup>9</sup> a alors été développée, dans laquelle le pro-

---

8. Critical Ray Scan Matching

9. Linear Approximation for Graph Optimization

cessus d'optimisation ne nécessite aucune estimation initiale.

En supposant que la position et l'orientation relatives sont indépendantes pour chaque nœud du graphe, un système d'équations équivalent à la fonction de coût non convexe est résolu. À cette fin, un ensemble de procédures basées sur la théorie des graphes permet d'obtenir une approximation de premier ordre du système non linéaire, au moyen d'une orientation linéaire et d'une estimation de position linéaire.

#### 7.3.3.4 HectorSLAM

HectorSLAM combine un système SLAM 2D basé sur une correspondance robuste des scans et une technique de navigation 3D utilisant un système de détection inertielle[8]. Le mouvement du robot est estimé en temps réel, en utilisant une fréquence élevée de mise à jour des données laser et un faible bruit de mesure de distance des LiDAR modernes. Les informations odométriques ne sont pas utilisées, ce qui donne la possibilité de mettre en œuvre cette approche pour des robots aériens comme un drone quadricoptère ou pour des robots terrestres opérant sur des terrains accidentés. D'un autre côté, cette approche peut être problématique lorsque les balayages laser sont peu fréquents et lorsque les estimations odométriques sont assez précises. En effet, elle n'utilisera pas les données odométriques pour corriger la position du robot.

L'estimation de la position 2D est basée sur l'optimisation de l'alignement des extrémités du faisceau laser avec la carte obtenue jusqu'à présent. Les points d'extrémité sont projetés sur la carte réelle et les probabilités d'occupation sont estimées. La correspondance du balayage est résolue à l'aide d'une équation de Gauss-Newton, qui trouve la transformation qui ajuste le mieux les faisceaux laser avec la carte. En outre, une représentation multi-résolution de la carte est utilisée pour éviter de rester coincé dans des minima locaux. Enfin, l'estimation de l'état 3D pour le filtre de navigation est basée sur l'EKF[9]. Cependant, cela n'est nécessaire que lorsqu'une unité de mesure inertielle (IMU) est présente, comme dans le cas des robots aériens.

#### 7.3.3.5 KartoSLAM

KartoSLAM est une approche SLAM basée sur les graphes développée par Karto Robotics de SRI International, qui a été étendue pour ROS. Un algorithme SLAM basé sur les graphes représente la carte au moyen de graphes. Dans ce cas, chaque nœud représente une position du robot le long de sa trajectoire et un ensemble de mesures de capteurs. Ceux-ci sont reliés par des arcs qui représentent le mouvement entre les positions successives. Pour chaque nouveau nœud, la carte est calculée en trouvant la configuration spatiale des nœuds qui sont compatibles avec les contraintes des arcs.

Dans la version de KartoSLAM disponible pour ROS, le solveur Sparse Pose Adjustment (SPA) est responsable à la fois de la correspondance des scans et des procédures de fermeture de boucle[9]. Plus le nombre de points de repère est élevé, plus la quantité de mémoire nécessaire est importante. Cependant, les algorithmes SLAM basés sur les graphes sont généralement plus efficaces que les autres approches lorsqu'il s'agit de maintenir une carte d'un environnement à grande échelle. Dans le cas particulier de KartoSLAM, il est extrêmement efficace, puisqu'il ne maintient qu'un graphe de positions.

### 7.3.3.6 GMapping

Gmapping est une approche SLAM par filtre particulaire Rao-Blackwellized basé sur des données laser[9]. Il s'agit de l'algorithme SLAM le plus largement utilisé dans les robots mobiles du monde entier. La famille d'algorithmes des filtres particulaires nécessite généralement un nombre élevé de particules pour obtenir de bons résultats, ce qui augmente sa complexité de calcul. De plus, le problème d'épuisement des particules associé au processus de rééchantillonnage du filtre particulaire diminue la précision de l'algorithme. Cela se produit parce que les poids d'importance des particules peuvent devenir insignifiants. Par conséquent, cela signifie qu'il existe une probabilité que des hypothèses correctes soient éliminées.

Une technique de rééchantillonnage adaptatif a été développée dans GMapping pour ROS, qui minimise le problème d'épuisement des particules, puisque ce processus n'est effectué que lorsqu'il est nécessaire. L'algorithme GMapping est conçu de telle manière qu'il intègre les données actuelles du LiDAR avec le modèle de mouvement de l'odométrie[8]. Cela permet de réduire l'incertitude sur la position du robot lors de l'étape de prédiction du filtre particulaire. En conséquence, le nombre de particules nécessaires est réduit puisque l'incertitude est plus faible, grâce au processus de correspondance des balayages.

### 7.3.3.7 Cartographer

La solution de Google pour le SLAM, appelée Cartographer, est un algorithme d'optimisation de graphe. C'est l'algorithme le plus complexe aujourd'hui de par ses multiples configurations possibles, mais aussi le plus puissant de par la qualité de ses résultats en temps réel.

Le code source ouvert de Google se compose de deux parties : Cartographer et Cartographer\_ROS. La fonction de Cartographer est de traiter les données du LiDAR, de l'IMU et des odomètres pour construire une carte. Cartographer\_ROS acquiert ensuite les données des capteurs via le mécanisme de communication ROS et les convertit au format Cartographer pour qu'elles soient traitées par Cartographer, tandis que le résultat du traitement Cartographer est libéré pour être affiché ou stocké.

Cartographer possède un fichier de configuration et de lancement individuel. De nombreux paramètres peuvent être réglés dans le fichier de configuration tels que la distance maximale du laser, la mise à jour angulaire et linéaire de la position du robot, le ratio d'échantillonnage des données laser, la période de publication de la carte, etc.

La principale différence par rapport à Gmapping et HectorSLAM est l'utilisation d'une IMU. Cependant, même si Cartographer est conçu pour travailler avec des données IMU, il peut également travailler avec l'odométrie.

**Remarque** C'est d'ailleurs l'odométrie qui a été utilisée pour comparer les performances des trois algorithmes retenus dans la partie suivante dans un souci de fidélité à la configuration matérielle du robot Kompaï.

### 7.3.3.8 Bilan

Après comparaison et mise en commun des résultats expérimentaux décrits dans les articles [10][9][8][6][4][7], la Figure 7.9 donne le classement global et détaillé des algorithmes ROS selon des critères définis.

Global ranking		1	2	3	4	5	6	7	8
SLAM algorithm		Cartographer	GMapping	Karto	Hector	LAGO	Core	Tiny	CRSM
Classification		Pose-graph optimization	Bayes filter	Pose-graph optimization	Bayes filter	Pose-graph optimization	Bayes filter	Bayes filter	Bayes filter
Based on		/	Fast SLAM (RBPF)	/	EKF	/	Tiny SLAM (Particle filter)	Particle filter	Particle filter
Input data		2D laser scan							
Output data		2D occupancy grid							
Odometry needed		YES or IMU	YES	YES	NO	YES	YES	YES	YES
Loop Closure		YES	YES	YES	YES	YES	NO	NO	NO
Evaluation criteria (ranking from 1 to n)	CMSE		1		2				3
	SDS		2		1				3
	SSIM		1	3	2				
	Proportion of occupied space 1		3	1	2	4	5		
	CPU load (%)		3	1	4	5	2		
	Line thickness & map correspondance	1	2		3			4	
	Proportion of occupied space 2	1	2		3			4	
	Number of enclosed areas	3	2		1			4	
	Corner count	1	2		3				
	K-nearest neighbours	1	3		2				

FIGURE 7.9 – Comparaison des différentes méthodes SLAM 2D sous ROS

Les critères de comparaison utilisés sont les suivants :

- **CMSE** - *Corner Mean Square Error* : Variance sur l'erreur de placement des coins de l'espace cartographié (distance entre le coin réel et celui trouvé par l'algorithme SLAM). Plus cette variance est faible, plus la carte générée est fidèle à la réalité.
- **SDS** - *Summation of the Differences of the lengths of the Sides of the maps* : somme des différences de longueur des côtés entre la carte réelle et la carte générée par l'algorithme. Plus cette somme est faible, plus la carte générée est fidèle à la réalité.
- **SSIM** - *Structure Similarity Index Metric* : Indice métrique de similitude de structure entre la carte générée et la carte réelle (obstacles rencontrés, murs, etc.). Plus cet indice est élevé, plus la carte générée est fidèle à la réalité.

- **Proportion of space occupied** - Proportion de l'espace occupé par un obstacle ou un mur. Plus cette proportion est grande, moins il y a d'espace laissé comme "incertain" par l'algorithme, plus la carte est fidèle à la réalité.
- **CPU load** - Charge du processeur utilisée pour cartographier. Plus elle est faible, plus l'algorithme SLAM est performant.
- **Line thickness and map matching** - Épaisseur des murs et obstacles de la carte et correspondance entre la carte réelle et la carte générée. Plus l'épaisseur est fine, plus l'algorithme SLAM est précis sur la cartographie.
- **Number of enclosed areas** - Nombre de zones avec un contour fermé : obstacles détournés, et contour global de l'espace cartographié. Plus les obstacles et murs sont bien détournés (pas de zone laissée comme "incertaine"), plus l'algorithme est précis sur la cartographie.
- **Corner count** - Nombre de coins comptés sur la carte générée. Plus le nombre de coins est proche du nombre de coins réels, plus la carte générée est fidèle à la réalité. Des coins peuvent être comptés là où réellement il n'y en a pas et inversement si la cartographie est trop peu précise.
- **K-nearest neighbours** - Somme des distances entre chaque cellule occupée dans la carte réelle et la cellule occupée correspondante la plus proche dans la carte générée (recherche des k plus proches voisins). Plus cette somme est grande, plus la carte générée diverge de la carte réelle.

Suite à cette comparaison, il me faut tester les trois meilleurs algorithmes sur le robot Kompaï (Cartographer, GMapping et Karto), en s'adaptant à sa configuration matérielle (LiDAR et odométrie à disposition notamment).

## 7.4 Modifications du logiciel de cartographie *Slam2d*

Les modifications et inchangés par rapport à la version précédente sont :

- (*modif*) La création de 2 onglets : *Default* avec boutons *Connect*, *Start*, *Stop* et le choix du nom de la carte (voir Figure 7.10), *Advanced* avec le choix de l'adresse IP et le choix de la résolution (voir Figure 7.11). L'adresse 192.168.1.2, la résolution à 0.04m par case de la grille d'occupation, et le nom de la carte "map" sont toujours par défaut. L'onglet *Default* est destiné à un usage utilisateur basique, et l'onglet *Advanced* à un usage utilisateur expérimenté.
- (*modif*) Les boutons sont plus gros et tout le texte avec une taille plus grande.
- *textbf(modif)* L'extension ".png" dans le nom de la carte n'est plus obligatoire! Il faut simplement mentionner le nom de la carte, par exemple "map". Si l'utilisateur entre "map.png", cela fonctionne toujours. Le programme ajoute automatiquement l'extension ".png" uniquement si elle n'est pas déjà présente.
- (*modif*) Un recentrage automatique de l'affichage de la carte sur le robot au click sur *Start*.
- (*modif*) Le bug **Karto** n'apparaît plus. Pour qu'il ne se produise plus, un nouveau fichier PNG est créé toutes les 2 secondes pour afficher la carte mise à jour. Chaque fichier généré

est nommé en prenant le nom de la carte donné par l'utilisateur (par exemple "map") et en y ajoutant le numéro de la carte générée (par exemple "map\_5" pour la 5ème carte). Les anciens fichiers sont supprimés au fur et à mesure. Cependant, il se peut qu'à l'appui sur Stop ou en quittant l'application **slam2d**, plusieurs PNG demeurent dans le dossier. Il faut alors prendre le fichier le plus récent, soit celui dont le numéro est le plus grand : le dernier nom de carte générée est visible sur l'IHM dans le *TextInput* du nom de la carte.

- (*modif*) A la fermeture de l'application **slam2d**, envoi d'une requête *HTTP PUT abort* pour rendre la main à **KomNAV** sur la JETSON et ainsi visualiser la position du robot et son laser sur le dashboard (sans avoir à cliquer sur "Abort" dans l'onglet "Navigation").
- (*modif*) Ajout d'une pop-up avec mot de passe pour accéder à l'onglet *Advanced*. Le mot de passe à saisir en clair est "user123".
- (*inchangé*) Il est toujours possible de renommer la carte pendant que le processus de cartographie tourne mais le délai de 2 secondes laisse peu de temps à l'utilisateur pour réaliser cette action... Dans ce cas, si on veut renommer la carte, on renomme directement le dernier fichier PNG après avoir appuyé sur *Stop*.
- (*inchangé*) On peut bien sûr toujours enchaîner les cartographies sans quitter l'application **slam2d**. Il suffit d'appuyer sur *Stop*, de donner un nouveau nom à la carte qui va être créée et d'appuyer à nouveau sur *Start*.

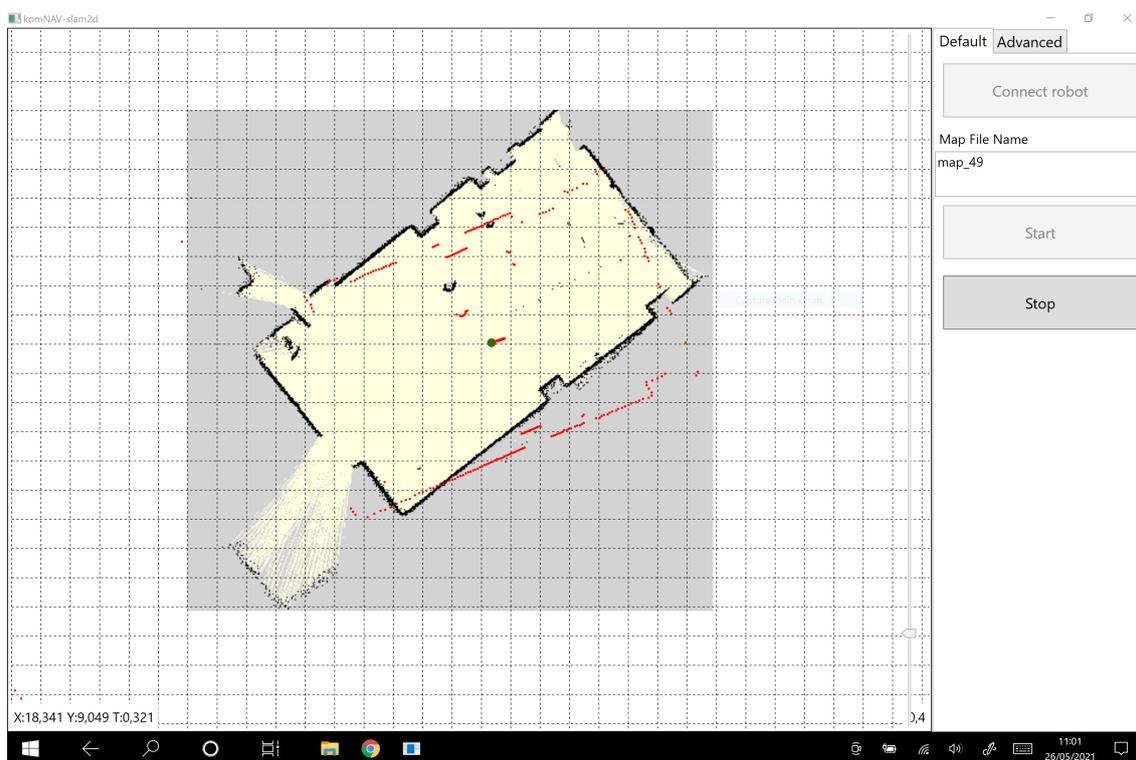
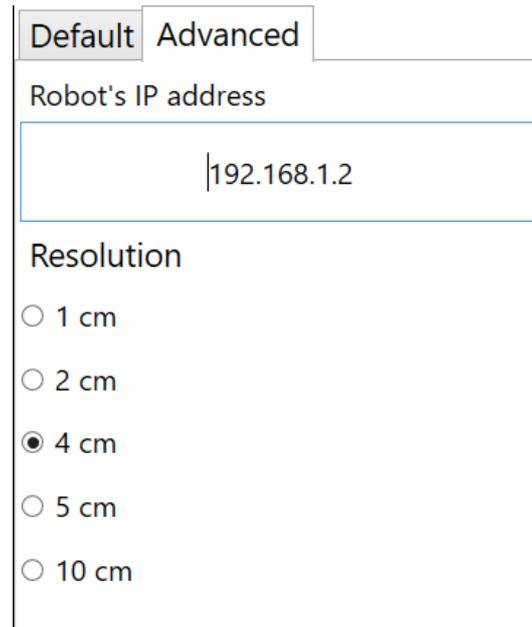


FIGURE 7.10 – Nouvel onglet *Default* dans le logiciel **Slam2d**

FIGURE 7.11 – Nouvel onglet *Advanced* dans le logiciel **Slam2d**

## 7.5 Veille technologique sur le système RTLS de Terabee

Fonctionnement du RTLS<sup>10</sup> Terabee :

- Utilisation pour une localisation :
  - **2D** : 3 balises émettrices nécessaires pour triangulariser la position du robot + 1 balise réceptrice sur le robot
  - **3D** : 4 balises émettrices nécessaires pour triangulariser la position du robot + 1 balise réceptrice sur le robot
- Technologie utilisée pour les balises émettrices : **UWB<sup>11</sup> à 6.5 GHz** (channel 5 de l'UWB). Pas d'interférence avec les réseaux Wi-Fi 2.4GHz et 5.8GHz (communément appelé 5GHz). Peu de réflexion donc bonne dissipation des signaux lorsqu'ils rencontrent des obstacles, ce qui minimise le bruit sur la position calculée du robot.
- Fréquence maximale de relocalisation : **10Hz**
- Portée maximale du signal : **40m**
- Alimentation balise émettrice ou réceptrice : **5 ou 24V** avec une consommation maximale de 25mA en 24V et 120mA en 5V.
- Précision des données de localisation : + **ou - 10cm** dans le plan (axes x et y), + **ou - 30cm** en hauteur (axe z). On n'est donc pas sur la précision d'1 cm annoncée initialement.
- Interface de communication : **RS232**
- Petit et léger : **Ø66 mm x 62.5 mm / 45g**

10. Real Time Localization System

11. Ultra-WideBand

- Toutes balises peuvent être utilisées soit en mode récepteur, soit en mode émetteur. Les balises émettrices sont espacées d'environ **20m**, ce qui permet bien de couvrir **1600m<sup>2</sup>** avec 4 balises disposées à chacun des coins d'un carré de 20m de côté.

Le système de localisation en temps réel (RTLS) de Terabee pourrait être utilisé à la place de l'odométrie qu'on utilise actuellement sur le kompaï pour le SLAM. Il s'agit en effet d'un système de balises actives émettrices et d'un récepteur sur le robot mobile qui **triangularise** la position du robot. Le RTLS de Terabee possède un système de positionnement automatique des balises émettrices dans l'espace pour faciliter son déploiement.

Cependant, l'intérêt est amoindri tant qu'il s'agit de robotique mobile au sol. L'odométrie actuelle du robot Kompaï est suffisamment précise pour que le robot soit correctement localisé.

En revanche, des balises passives (ou ancres) pourraient être placées à certains endroits pour améliorer encore plus la précision sur la position du robot, et ainsi recalibrer l'odométrie.

Sur un drone, le système peut être intéressant, surtout si l'IMU est de mauvaise qualité. En intérieur, les drones se basent principalement sur leur IMU (capteur proprioceptif) pour calculer leur position, donc l'utilisation de balises passives ou actives (capteurs extéroceptifs) est quasi indispensable. Sachant que si on considère un espace fermé sombre, il est plus malin d'utiliser des balises actives.

Par contre, le système de Terabee est excessivement performant et coûteux s'il s'agit d'un espace intérieur petit comme une salle de cinéma. En effet, il faudrait 4 balises émettrices par salle pour faire une localisation 3D du drone, sachant que leur portée maximale est de 40m.

**Remarque** Le tracker ou balise réceptrice est petite et légère donc pourrait être positionnée sur un drone. Mais aujourd'hui, Terabee propose une solution déjà clé en main [ici](#).

## 7.6 Organigrammes logiciels, matériels et réseaux

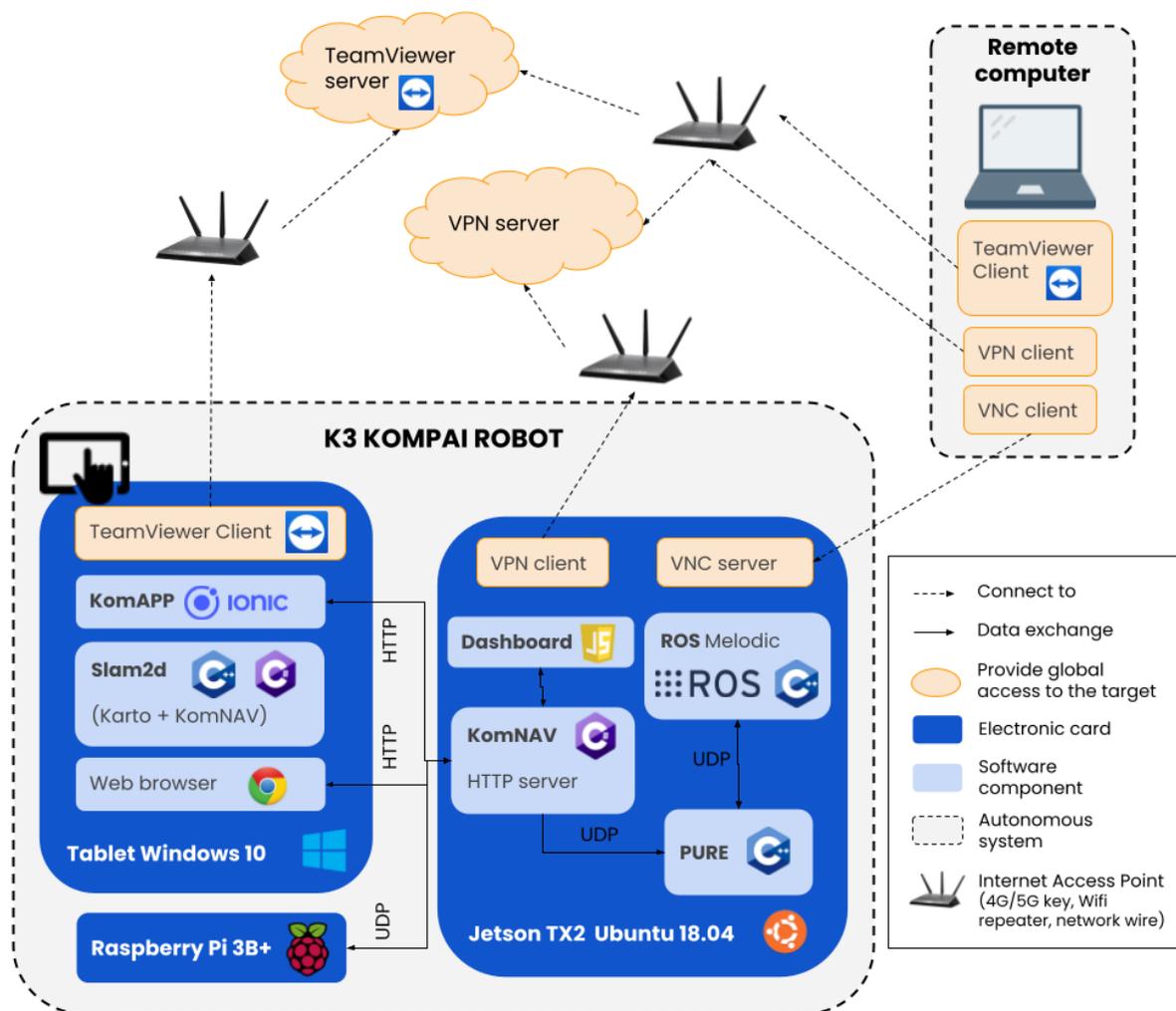


FIGURE 7.12 – Communications réseaux possibles sur le robot Kompaï

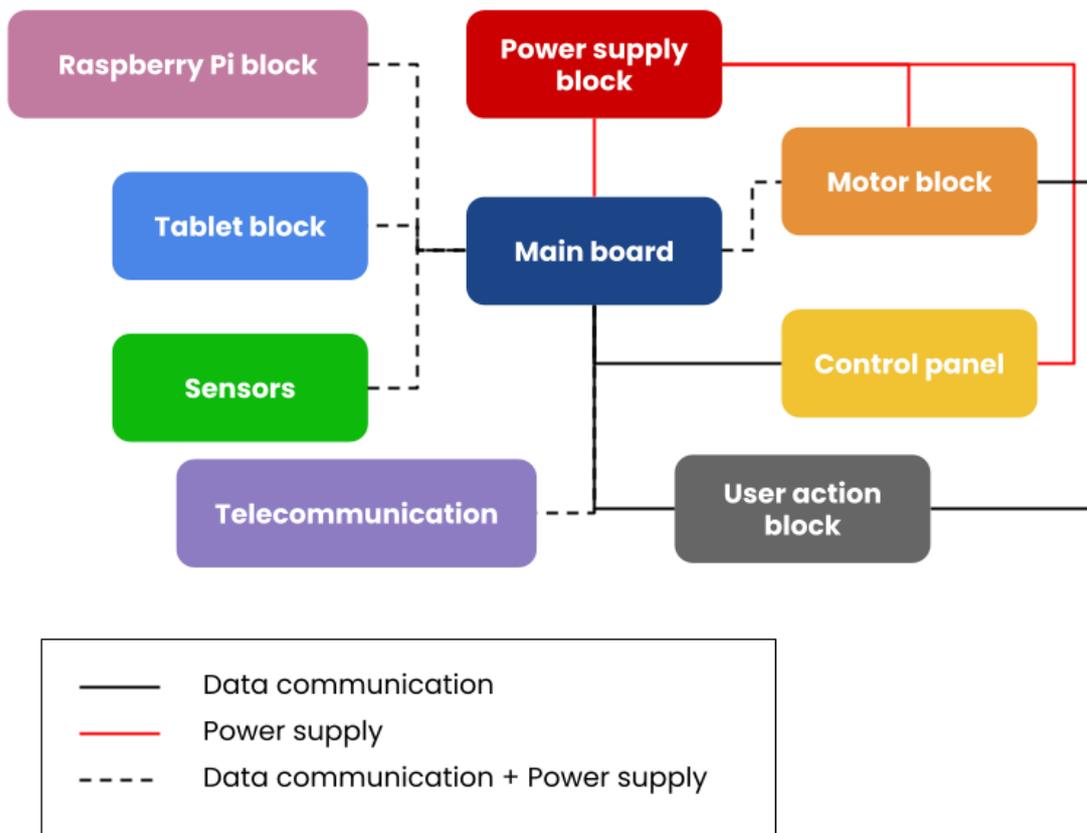


FIGURE 7.13 – Architecture matérielle globale du robot Kompaï

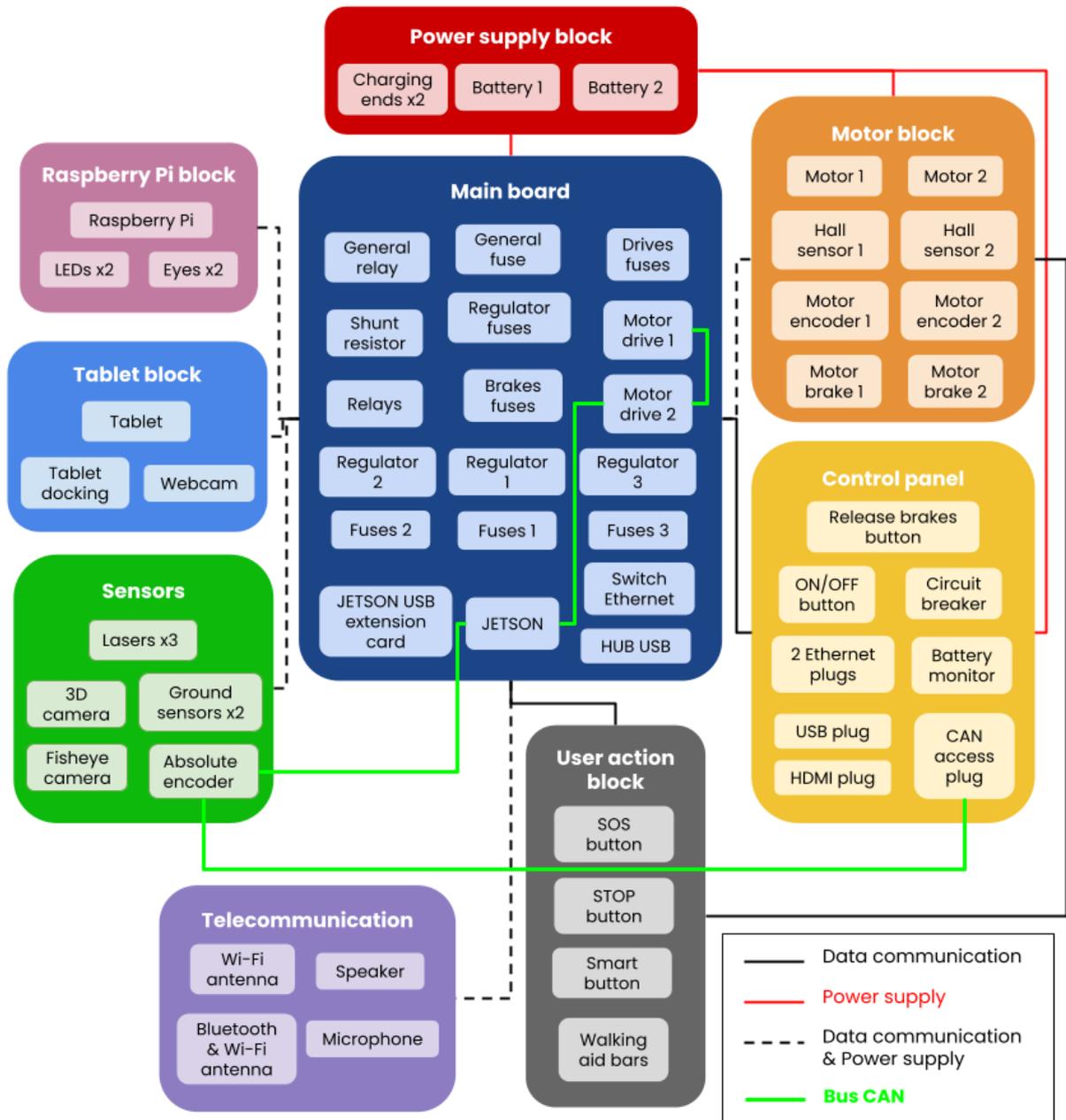


FIGURE 7.14 – Architecture matérielle globale détaillée du robot Kompaï

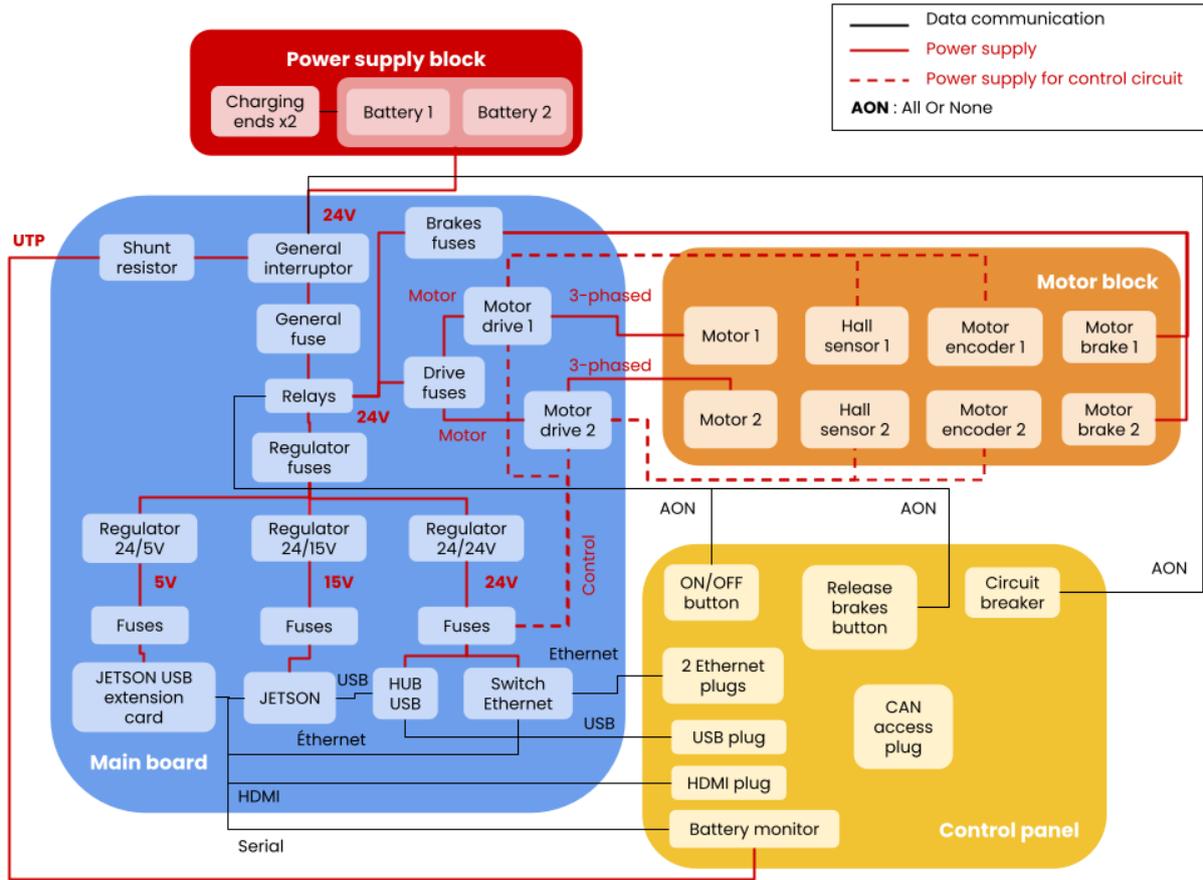


FIGURE 7.15 – Architecture matérielle détaillée des moteurs, du panneau de configuration et de l’alimentation du Kompaï

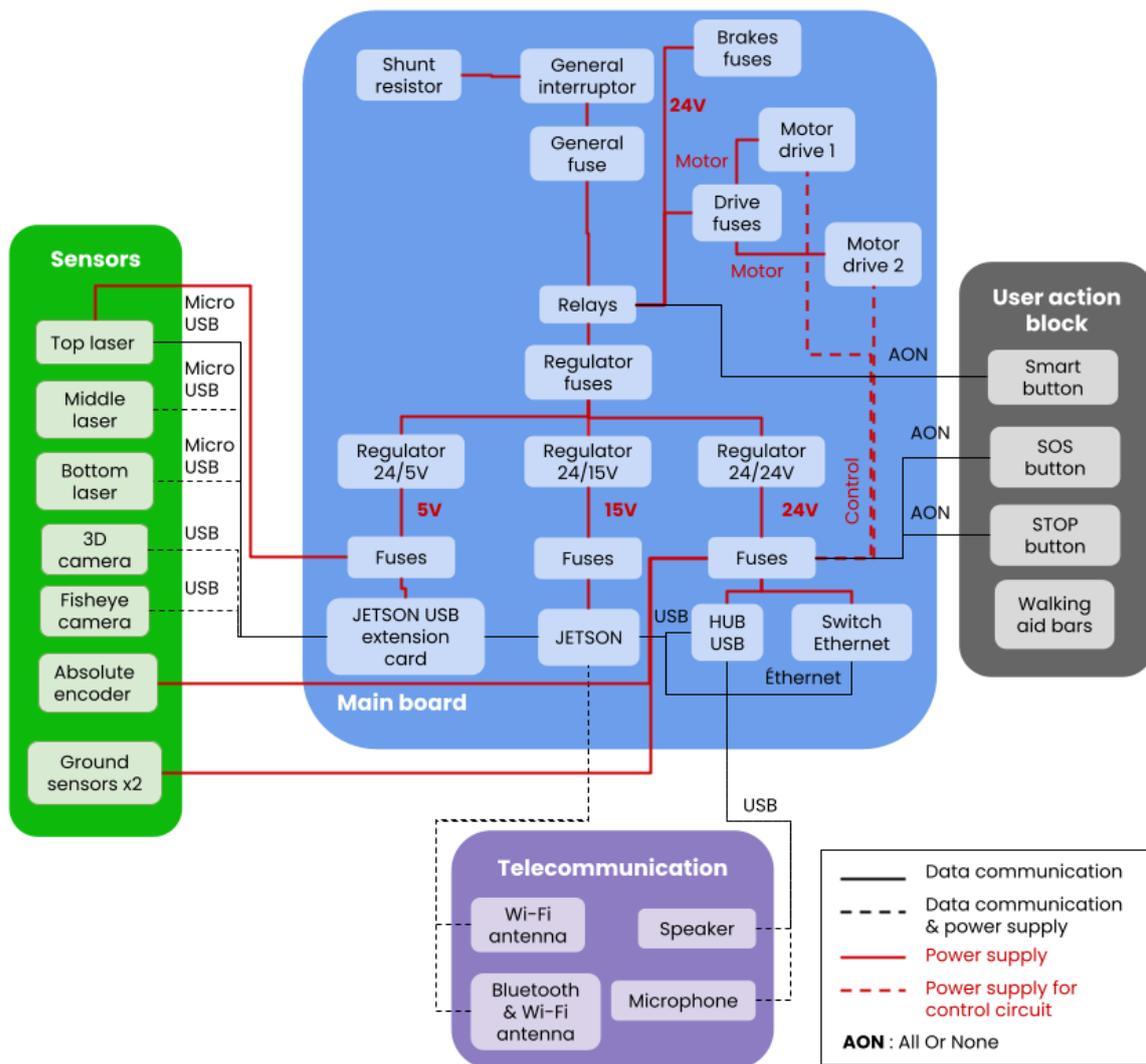


FIGURE 7.16 – Architecture matérielle détaillée des capteurs, des télécommunications et des actions utilisateur du Kompai

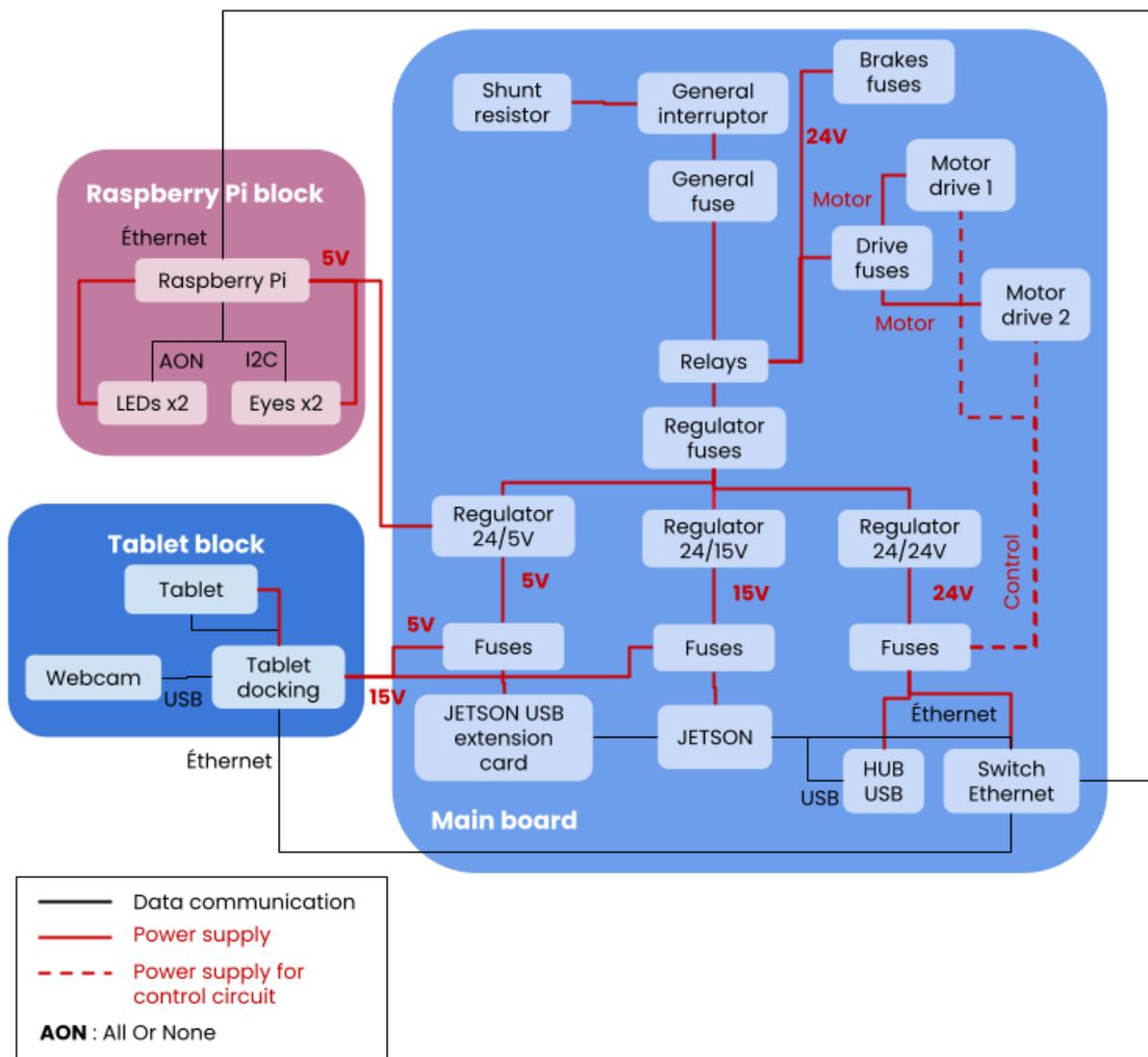


FIGURE 7.17 – Architecture matérielle détaillée de la tablette et de la Raspberry Pi du Kompaï

## 7.7 ROS node graph

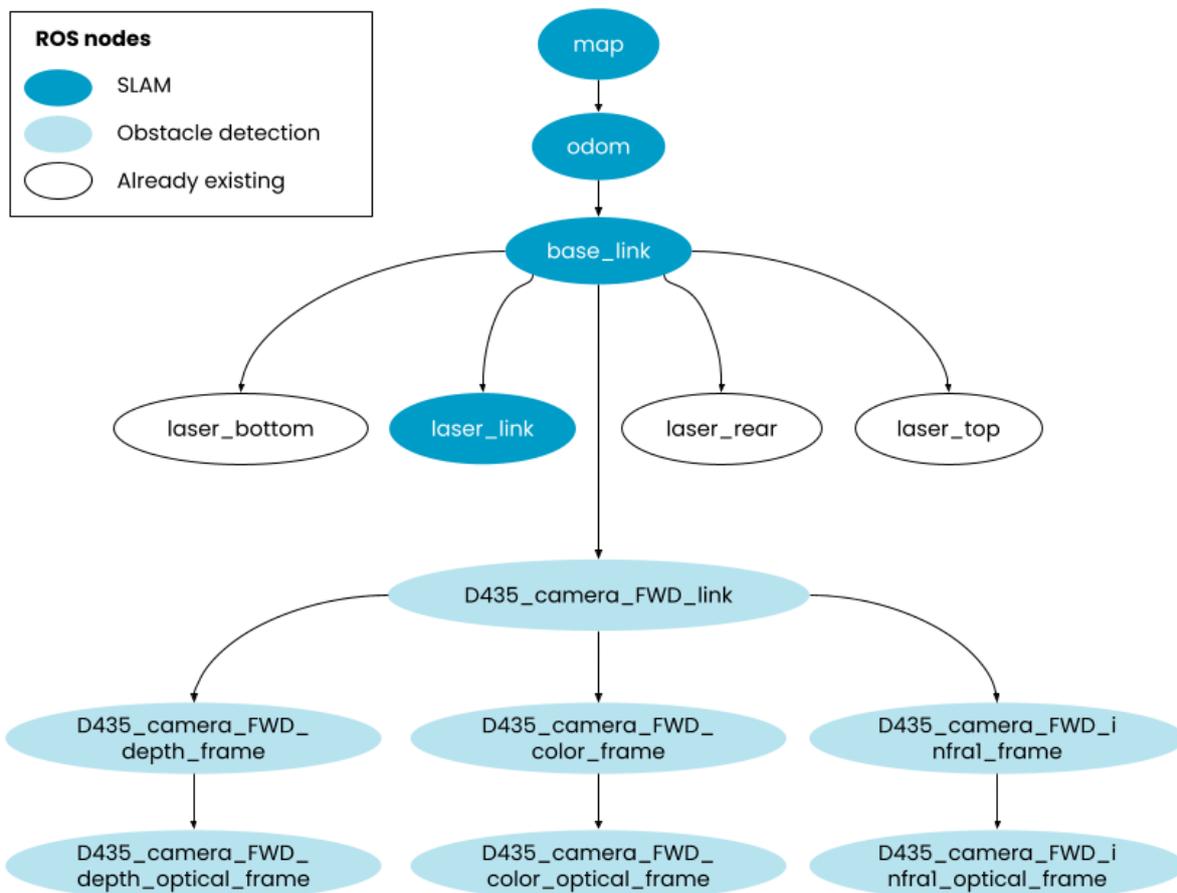


FIGURE 7.18 – ROS node graph après intégration des noeuds de détection d’obstacles et de cartographie avec PURE



## RAPPORT D'EVALUATION ASSESSMENT REPORT

**Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :**  
**At the end of the internship, please return this report via mail or email to:**

ENSTA Bretagne – Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE  
☎.33 (0) 2.98.34.87.70 / [stages@ensta-bretagne.fr](mailto:stages@ensta-bretagne.fr)

### I - ORGANISME / HOST ORGANISATION

NOM / Name KOMPAL ROBOTICS

Adresse / Address 97 Allée Théodore Monod, 64210 Bidart

Tél / Phone (including country and area code) +33 (0)6 62 26 34 43

Nom du superviseur / Name of internship supervisor

TRIAUX ANTOINE

Fonction / Function Référent capteurs

Adresse e-mail / E-mail address antoine.triaux@kompai.com

Nom du stagiaire accueilli / Name of intern

**ESTELLE ARRICAU**

### II - EVALUATION / ASSESSMENT

Veuillez attribuer une note, en encrant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**  
*Please attribute a mark from A (excellent) to F (very weak).*

#### MISSION / TASK

◆ La mission de départ a-t-elle été remplie ? A B C D E F  
*Was the initial contract carried out to your satisfaction?*

◆ Manquait-il au stagiaire des connaissances ?  oui/yes  non/no  
*Was the intern lacking skills?*

Si oui, lesquelles ? / If so, which skills? \_\_\_\_\_

#### ESPRIT D'EQUIPE / TEAM SPIRIT

◆ Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work) A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / If you wish to comment or make a suggestion, please do so here \_\_\_\_\_

## COMPORTEMENT AU TRAVAIL / BEHAVIOUR TOWARDS WORK

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?

*Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)?*

A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* \_\_\_\_\_

---

## INITIATIVE – AUTONOMIE / INITIATIVE – AUTONOMY

Le stagiaire s'est-il rapidement adapté à de nouvelles situations ?

A B C D E F

(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

*Did the intern adapt well to new situations?*

*(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)*

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* \_\_\_\_\_

---

## CULTUREL – COMMUNICATION / CULTURAL – COMMUNICATION

Le stagiaire était-il ouvert, d'une manière générale, à la communication ?

A B C D E F

*Was the intern open to listening and expressing himself /herself?*

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* \_\_\_\_\_

---

## OPINION GLOBALE / OVERALL ASSESSMENT

◆ La valeur technique du stagiaire était :

A B C D E F

*Please evaluate the technical skills of the intern:*

## III - PARTENARIAT FUTUR / FUTURE PARTNERSHIP

◆ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?

*Would you be willing to host another intern next year?*  oui/yes

non/no

Fait à Bidart , le 30/08/2021

Signature Entreprise ANTOINE TRIAUX

Signature stagiaire

*Merci pour votre coopération*  
*We thank you very much for your cooperation*