

RAPPORT DE PROJET



11/04/2016

LittleThumb : Un robot autonome

Une solution simple et efficace permettant à un robot de téléprésence d'être piloté et de retourner de façon autonome à sa base.

Rapport de projet

LITTLETHUMB : UN ROBOT AUTONOME

Table des matières

REMERCIEMENT	2
INTRODUCTION.....	2
PARTENAIRES ET PARTICIPANTS	3
AXYN	3
Epitech.....	3
Pixmap	3
SmarDTV	4
1. ÉTUDE DU PROJET	5
1.1 Contraintes et situations initiales	5
1.2 Pourquoi LittleThumb.....	5
1.3 Élaboration du cahier des charges.....	6
1.4 Objectifs à réaliser.....	6
2. ÉLABORATION DU PROJET	7
2.1 Les différentes phases du projet.....	7
2.1.1 Communication entre les partenaires.....	7
2.1.2 Architecture du projet	7
2.1.2 Prise en main des outils.....	8
2.1.3 Création des fonctions principales.....	8
2.1.3.1 Serveur d'interfaçage	8
2.1.3.2 Interface de contrôle client	9
2.1.3.3 Simulateur de robot et d'environnement.....	10
2.1.3.4 Analyse de l'environnement.....	11
2.2 La répartition du travail	11
2.3 Les problèmes et les solutions envisagées	12
3. RESULTATS ET PERSPECTIVES	14
3.1 État final du projet.....	14
CONCLUSION	15

REMERCIEMENT

Nous tenons à remercier SMARDTV qui nous a été d'une aide précieuse, nous a prêté la tablette Tango et nous a consacré du temps pour le développement du projet. Nous remercions AXYN pour nous avoir fourni le robot sans lequel nous n'aurions pas pu tester nos développements. Nous remercions PIXMAP pour leur apport innovant sur la modélisation 3D, pour nous avoir formés sur le middleware ROS et le SDK PIXMAP et de nous avoir fourni des accès à leur API PIXMAP.

INTRODUCTION

Le projet LittleThumb consiste à développer une application permettant de piloter un robot sur Smartphone, tablette, ordinateur ou tout autre périphérique standard. Il faut qu'il puisse de lui-même, c'est-à-dire sans interaction d'utilisateurs, par navigation automatique (mode « patrouille »), retourner à sa position de départ ou à sa base de chargement enregistrée.

Le robot de téléprésence serait pour un usage aussi bien à domicile qu'en entreprise et serait capable de refaire n'importe quel parcours effectué.

Inscrit dans le cadre d'un module de partenariat et d'innovation, le projet proposé par SmarDTV nous a semblé être une évidence pour mettre en pratique toutes nos connaissances acquises sur un projet complexe et concret. Nous avons choisi de réaliser l'application permettant le contrôle complet d'un robot de téléprésence. De notre point de vue, ce projet présente trois caractéristiques fondamentales :

- ❖ D'une part la nouveauté de la tablette Tango sur laquelle va être déployée l'application de pilotage,
- ❖ D'autre part le system ROS qui nous permet d'avoir un premier pied dans la "vraie" robotique,
- ❖ Enfin la logique algorithmique appliquée à des contraintes réelles.

C'est ainsi que nous avons essayé de réaliser un ensemble d'applications fonctionnelles, remplissant les conditions imposées par le cahier des charges établi en collaboration avec les principaux utilisateurs. La nouveauté réside notamment dans la manière de concevoir et de développer. Après avoir présenté les différents intervenants, nous verrons la présentation du projet et ses principaux objectifs, son élaboration. Enfin, nous verrons les résultats obtenus ainsi que les évolutions possibles du projet et de l'application.

PARTENAIRES ET PARTICIPANTS

L'équipe du projet est constituée au final de quatre entreprises en collaboration, c'est-à-dire, AXYN, Epitech, SmarDTV et PIXMAP.

AXYN

Axyn Robotique est une startup spécialisée dans la robotique dont l'objectif est la conception de robots d'aide aux personnes à mobilité réduite.

Dans le cadre du projet, Frank ANJEAUX nous a fourni la base du robot, nous a présenté l'architecture matérielle du robot et nous a formé sur les notions de contrôleur PID, de filtre de Kalman, ainsi que le développement UBYX.



Robot Ubyx

Epitech

Epitech est une école informatique qui forme en 5 ans après-bac des experts du secteur. Elle permet de transformer une passion en expertise via une pédagogie résolument innovante, organisée par projets.

Dans le cadre du projet, une équipe de 5 étudiants en 3e année (Cintia FERREIRA, Flavien MAILLOT, Julien ATHOMAS, Sophie DUMONT, Timothy BURDAIRON) ont apporté leurs connaissances techniques pour développer les applications associées au LittleThumb.

Epitech, avec Fabien REYNAUD, a aussi été chargé des communiqués de presse ainsi que de la communication générale sur le projet.

Pixmap

Pixmap a développé un programme permettant de représenter l'environnement d'un robot en 3D, d'une manière photoréaliste, lui permettant de se localiser, quelle que soit sa forme, taille ou le mode de locomotion, le tout en temps réel.

Concernant le projet, PIXMAP a abordé les concepts de robotique mobile et de navigation, il nous a introduit sur le middleware ROS et nous a présenté leur SDK PIXMAP. De plus PIXMAP a mis à notre disposition des accès à l'API PIXMAP.

SmarDTV

SmarDTV est un leader international dans les technologies de télévision payante. Membre fondateur du CI Plus Forum, SmarDTV conçoit et fabrique des modules hautement sécurisés et interchangeables qui offrent une alternative attrayante aux décodeurs pour accéder à la télévision payante.

SmartDTV encadre le projet LittleThumb, elle apporte ses connaissances techniques avec trois référents (Brunot TONELLI, Marc LAMBERT, Olivier HUET), fourni la tablette Tango qui est le périphérique contrôlant le robot ainsi qu'une formation sur son SDK. C'est également notre interlocuteur privilégié pour le suivi du projet.



Tablette Tango recto-verso

1. ÉTUDE DU PROJET

1.1 Contraintes et situations initiales

En matière de communication à distance et de collaboration SmarDTV possède notamment un robot de téléprésence basique. L'inconvénient, il n'a pas de sécurité pour les grands vides tels que les marches, il ne retourne pas à sa base de façon autonome, enfin il est fabriqué d'un bloc unique et non modulable. Côté pilotage, l'ergonomie, le style visuel ainsi que le moteur de développement de l'application commence à "vieillir".

C'est pourquoi SmarDTV a voulu s'engager sur le créneau de la téléprésence en entreprise et à domicile en lançant un partenariat avec AXYN, PIXMAP et Epitech. Le sujet consiste à développer une application pour un robot de téléprésence basée sur le système ROS permettant la communication entre le robot, la tablette et le client de contrôle. Le système doit permettre le retour du robot de façon autonome à sa base, se localiser dans un environnement inconnu, si possible avoir une gestion des chemins les plus courts ou encore un système avancé sur la gestion de l'autonomie du robot.

1.2 Pourquoi LittleThumb

Peu modulables et onéreux, les robots de téléprésence montrent des limites quant à leur système de navigation. Pourtant, de plus en plus utilisé dans les foyers et les entreprises, il devient nécessaire de pouvoir créer un robot modulable et le plus autonome possible.

LittleThumb est un projet qui a été développé dans l'esprit de faciliter la conception du robot, alléger ses coûts sans pour autant réduire la couverture fonctionnelle et simplifier son utilisation (tant en pilotage qu'en mode autonome).

Pour atteindre l'objectif de réduction des coûts, plusieurs moyens sont mis en place :

Tout d'abord, LittleThumb prévoit un programme exportable sur tout device de contrôle Android (ici une tablette Tango), ce qui induit que le device peut être changé pour un périphérique différent. Le client de contrôle utilisateur doit être cross-platform permettant de ne pas avoir à développer des mises à jour sur plusieurs systèmes différents. Le système de communication utilise un standard dans le domaine de la robotique permettant de changer aisément la base du robot.

Ensuite sur les pièces elles-mêmes (la base, le device de contrôle), elles peuvent indifféremment être changées du moment qu'elles respectent les normes, les standards.

Enfin notre découpage en modules ou blocs des tâches à effectuer permet une maintenance plus simple du code.

Dans le cadre du projet, nous avons créé un module de retour autonome du robot, qui sera activé uniquement lorsque l'utilisateur libère le contrôle. Ce module terminé et fonctionnel, peut être au choix retravaillé ultérieurement ou laissé en l'état actuel. Il en va de même pour le serveur, le client de contrôle, le simulateur, etc.

Étudiés depuis la première année à Epitech, nous savons mettre en place de puissants algorithmes, permettant de répondre à des attentes diverses concernant le robot.

1.3 Élaboration du cahier des charges

Le cahier des charges a été réalisé à partir des contraintes rencontrées avec l'utilisation du robot de téléconférence possédé par SmarDTV. La rédaction du cahier des charges est orientée à travers deux principaux objectifs :

Premièrement, être en mesure de déplacer le robot où l'on souhaite à partir de l'interface de contrôle client.

Deuxièmement, permettre au robot de refaire un parcours en sens inverse de façon autonome.

Seulement ces étapes validées, il était envisagé de faire retourner le robot à sa base via des contraintes plus complexes telles que celles liées à l'environnement ou au chemin le plus court.

Pour remplir au mieux ces deux conditions, l'ensemble des applications doivent être fonctionnelles, c'est-à-dire que les fonctionnalités doivent être ciblées et précises, de plus, toute la partie de contrôle client et de simulation doit être ergonomique, ce qui signifie que toutes les fonctionnalités doivent être accessibles rapidement et simplement. Le cahier des charges va s'axer sur les points suivants:

- ❖ Un serveur permettant la communication simple entre tous les systèmes (tablette, contrôle client, simulateur)
- ❖ Un système d'asservissement sur la tablette
- ❖ Un système de contrôle client
- ❖ Un simulateur d'environnement et de robot

L'accent a été mis sur l'aspect fonctionnel du projet plutôt que sur l'optimisation, en ce sens qu'il était toujours possible de revenir sur cet aspect une fois les fonctionnalités principales développées.

Au final, le cahier des charges se résume en 3 points fondamentaux que l'équipe de projet doit respecter :

- ❖ Réaliser une application simple d'utilisation pour le contrôle client et le simulateur.
- ❖ Avoir un système fonctionnant totalement sur ROS
- ❖ Avoir un robot qui se déplace en fonction des choix utilisateurs et qui sait refaire un parcours en sens inverse.

1.4 Objectifs à réaliser

Au préalable, avant de commencer le développement en lui-même, il nous a fallu nous documenter et "apprivoiser" l'environnement de travail (configuration homogène, utilisation, etc.), c'est-à-dire ROS et Tango.

Ensuite, il a fallu mettre en place le moyen de communication entre les différents systèmes.

Enfin, ces prérequis réalisés, les piliers du projet pouvaient être développés (contrôle client, asservissement par la tablette tango, simulateur).

2. ÉLABORATION DU PROJET

2.1 Les différentes phases du projet

Afin de mieux se répartir le travail, nous avons découpé le projet en plusieurs phases :

- ❖ Communication entre les partenaires.
- ❖ Architecture du projet
- ❖ Prise en main des outils.
- ❖ Création des fonctions principales.

2.1.1 Communication entre les partenaires

Après la réunion de présentation du projet, nous avons mis en place plusieurs moyens de communication inter-entreprises.

Tout d'abord, nous avons créé un compte Yammer pour le projet en substitution aux e-mails. Cette solution permet un meilleur suivi des discussions et des différents problèmes rencontrés lors du développement du projet et une meilleure diffusion des documents et comptes rendus.

Nous avons également utilisé un service de stockage et de partage de fichiers dans le Cloud, il s'agit de One Drive de Microsoft ainsi que le calendrier en ligne.

Des réunions hebdomadaires ont été programmées avec l'équipe de SmarDTV tous les lundis en visio-conférence sur Skype afin de discuter des avancements ou des problèmes rencontrés sur le projet.

Enfin, un répertoire de dépôt Git a été mis en place en interne afin de partager le code source développé.

2.1.2 Architecture du projet

Tout d'abord, nous avons dû définir les nouvelles notions liées au domaine de la robotique et se documenter sur les outils mis à disposition.

Rapidement, il a été nécessaire d'architecturer les informations trouvées dans les documentations et lors de nos recherches afin de mieux comprendre et d'affiner nos choix concernant les composants ou les technologies à privilégier.

Nous avons abordé les notions de téléprésence, de contrôleur PID (Proportional Integrative Derivative) qui correspond à un système de régulation des mesures ou encore de Filtre de Kalman, qui permet d'éliminer les résultats incohérents et ainsi de réduire la marge d'erreur des mesures effectuées.

Nous avons étudié les différents modes de déplacements existants, les différents moyens de localisation ainsi que les senseurs (capteurs) existants notamment ceux utilisés par la tablette TANGO et les modules proposés par PIXMAP.

PIXMAP propose une technologie de localisation (PX2M) qui cartographie en 3D un décor en temps réel. Il s'agit d'un SDK simple et modulable qui utilise une caméra Kinect.

Les recherches concernant la communication ou l'alimentation de la tablette et du robot ont permis de soulever des problématiques importantes auxquelles nous devons trouver des solutions.

2.1.2 Prise en main des outils

En attendant que nous ayons tous des environnements de développement fonctionnels et homogènes, SmarDTV a mis à notre disposition une machine virtuelle avec tous les outils nécessaires préconfigurés.

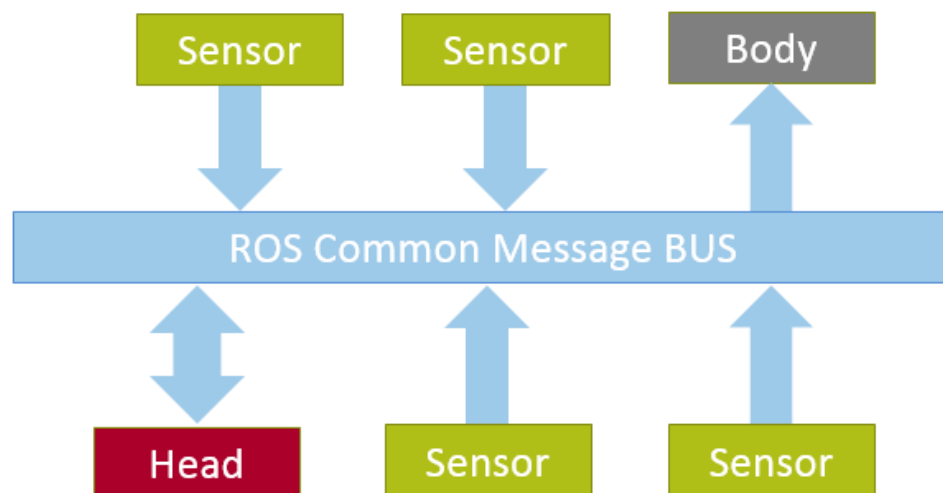
Ainsi nous avons pu découvrir les éléments fonctionnels comme l'API Tango, Rviz qui est le logiciel de visualisation de ROS et ROS.

2.1.3 Création des fonctions principales

2.1.3.1 SERVEUR D'INTERFAÇAGE

Le serveur ROS a été développé en C++. Il permet la communication entre l'interface de contrôle client, la tablette TANGO et le robot (ou simulateur).

Le serveur reçoit les commandes de l'interface de contrôle client et les transmet à la tablette TANGO qui les envoie au robot, puis la tablette récupère les données d'odométrie du robot et renvoie une réponse au serveur après traitement.

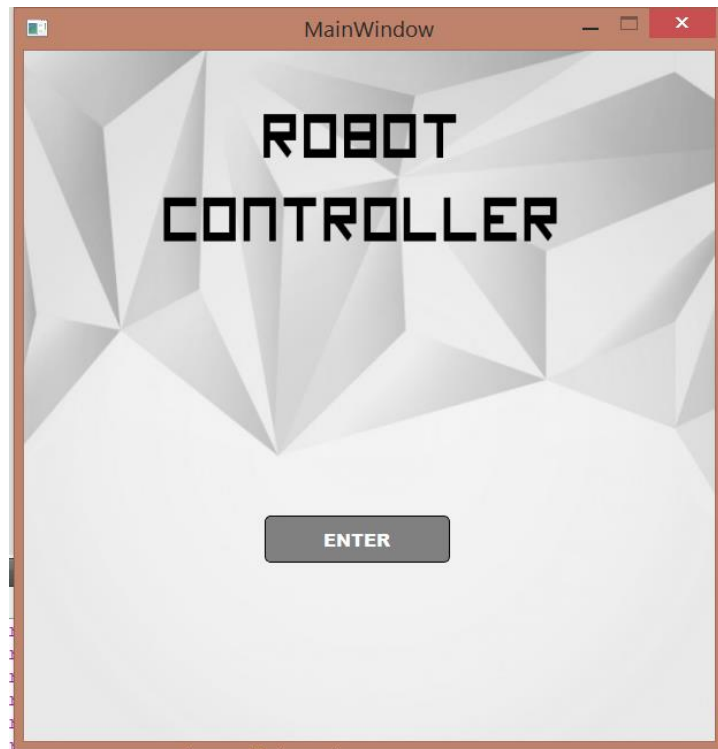


Communication ROS

Le serveur a été conçu de façon à être le plus abstrait possible afin d'être exporté potentiellement sous un autre système d'exploitation tel que Windows.

2.1.3.2 INTERFACE DE CONTROLE CLIENT

Il s'agit d'une interface graphique réalisée en C++ avec le Framework Qt qui permet de faire déplacer le robot via une interface ergonomique, simple et intuitive. De simples flèches permettent d'envoyer des instructions au serveur permettant de déplacer le robot.



Fenêtre de démarrage pour le contrôle client du robot

Les déplacements principaux matérialisés sont la marche avant, la marche arrière et les demi-tours. Nous avons ensuite ajouté deux fonctionnalités : la réalisation d'un trajet préenregistré et la fonction retour qui effectue le trajet réalisé en sens inverse.



Interface de contrôle utilisateur du robot

Le contrôleur client, dans une version future, permettra un retour visuel de la caméra du robot de façon à permet à l'utilisateur de diriger le robot sans l'avoir en vue. Il est également envisageable d'avoir des éléments supplémentaires sur l'interface permettant à l'utilisateur d'être renseigné sur des évènements importants tels qu'une collision imminente, une chute, une faible charge de la batterie, etc.

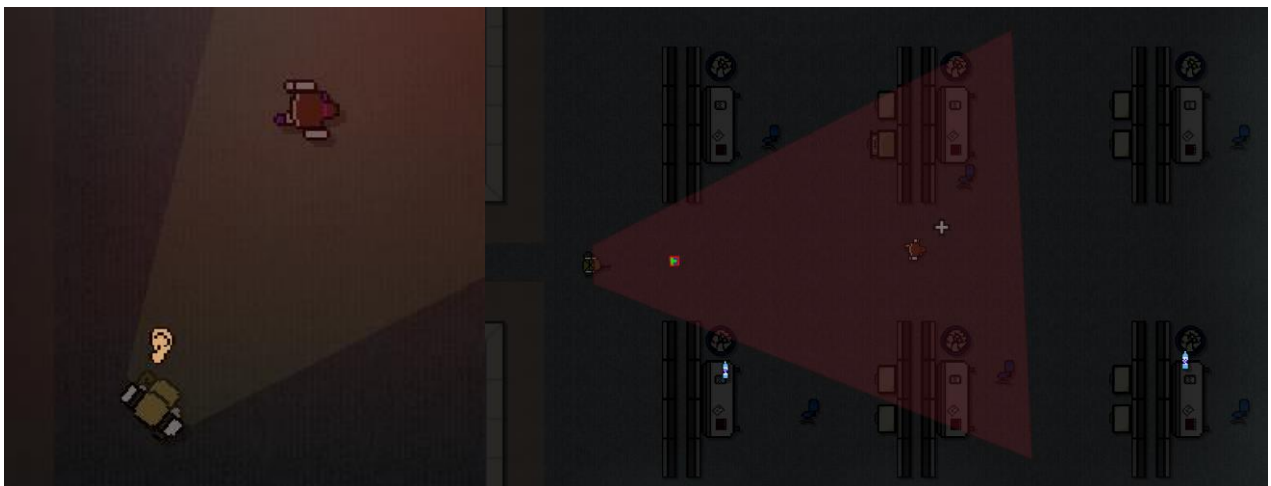
2.1.3.3 SIMULATEUR DE ROBOT ET D'ENVIRONNEMENT

Le simulateur développé en JAVA est composé de deux parties distinctes, le robot et la carte. Le simulateur de carte permet une génération d'environnements différents sur un système de tiled éditable via un logiciel spécialisé permettant de définir des zones de collision, de déplacement, de transition et de vide.



Configuration d'une carte d'intérieure en tiled

Le simulateur de robot peut être intégré dans le simulateur d'environnement, il peut se déplacer en fonction des instructions fournies par le contrôleur client ou de façon autonome via le mode patrouille et retourner des informations à la fois aléatoires concernant ses propres senseurs (roues par exemple) ou des détecteurs de collisions (un mur, un obstacle).



Détection de personnes (fig. gauche), Détection d'obstacle (fig. droite)

Ce simulateur possède une connexion socket pour la réception et l'émission des données.

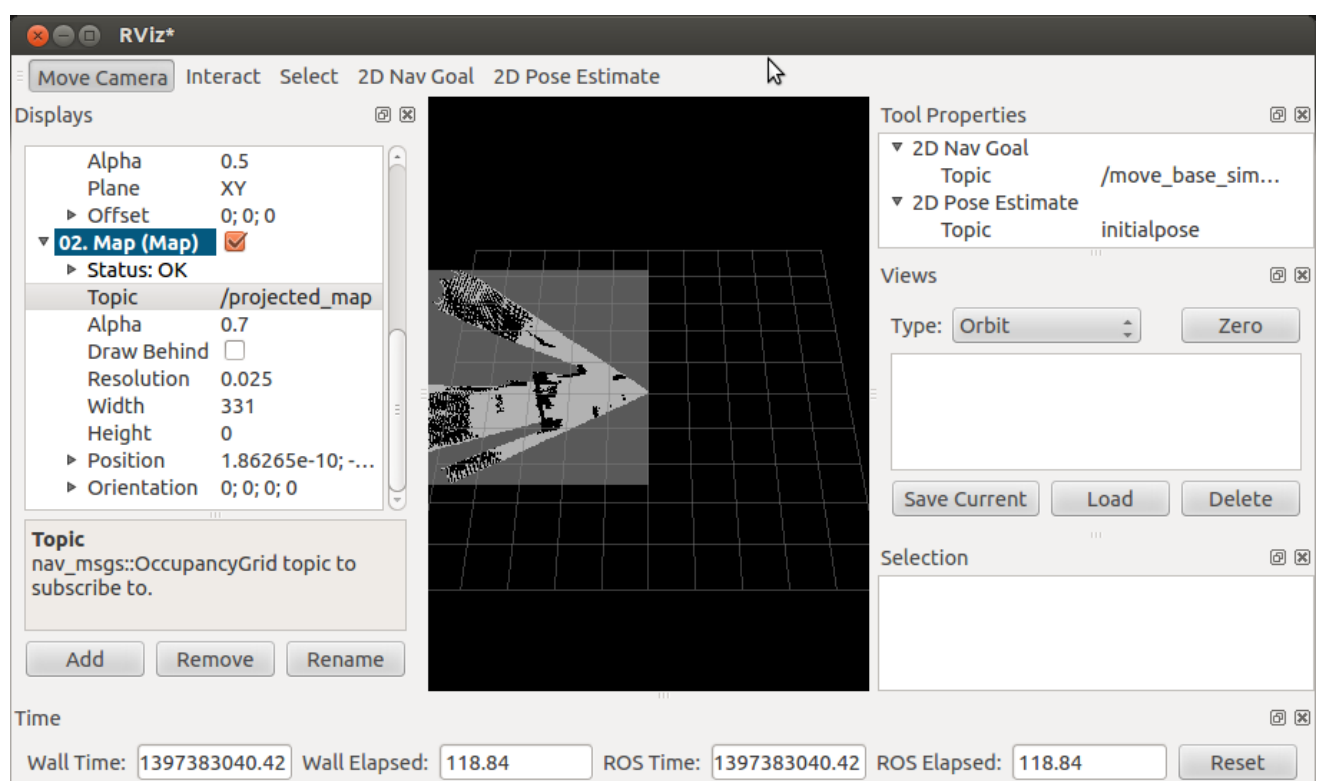
Le choix du langage JAVA du simulateur a été défini de tel sorte à ce qu'il soit facilement mis à jours et cross-plateforme.

Le simulateur nous a également permis de palier à l'absence du robot et de tester rapidement les avancées du serveur et de l'interface de contrôle client.

2.1.3.4 ANALYSE DE L'ENVIRONNEMENT

Afin de permettre le retour autonome du robot à sa base d'alimentation, il était nécessaire d'effectuer une analyse de l'environnement. Cette analyse est effectuée par des senseurs présents sur la tablette TANGO et directement exploitables sur ROS dans RVIZ.

Les données reçues permettent aussi bien de générer un nuage de points 2D ou 3D. Seules celles permettant de générer une grille d'occupation ont été retenues et utilisées. La grille permet en définitif d'établir d'une part une carte 2D de l'environnement d'autre part sur l'instant t de calculer la distance de collision avec un obstacle.



Grille d'occupation visualisée sur RViz

2.2 La répartition du travail

Pour permettre à chacun des membres de l'équipe de s'investir entièrement dans le projet, il était nécessaire d'effectuer une répartition des tâches selon l'envie et les compétences de chacun.

La répartition a été initialement prévue lors de la première réunion de groupe (avec SmarDTV).

Certaines tâches ont été affectées à certaines personnes d'autres ont été réalisées successivement par plusieurs personnes du groupe. Les comptes rendus des réunions ont ainsi suivis cette logique et ont été majoritairement fait par Sophie, Cintia.

Pour un aspect plus professionnel, il était nécessaire d'effectuer régulièrement la correction et la relecture des différents documents (documentations, comptes-rendus, etc), ce qui a été réalisé par Cintia.

Très rapidement, il s'est fait sentir le besoin d'avoir un outil de communication centralisé (Yammer), ainsi qu'un Drive pour mettre à disposition les différents documents. Le tout a été mis en place et configuré par Flavien.

Afin de garantir un bon suivi du projet par les partenaires, nous les avons associés à nos réunions hebdomadaires et avons mis en place la programmation de visio-conférences et de rendez-vous comme les sessions workshop, celles-ci étaient majoritairement organisées par Julien.

Ces bases établies nous avons enfin pu commencer à définir la répartition concernant le propre du projet. Les recherches pour l'architecture ont été faites par Sophie et Cintia. L'étude, la recherche et la documentation concernant PIXMAP a été faite par Timothy. Le serveur d'interfaçage entre la tablette TANGO et le robot ou le simulateur a été entièrement développé par Julien. La base du système d'asservissement (connexion sockets) a été développée par Flavien puis repris avec Julien pour incorporer le système ROS sur lequel Timothy et Cintia ont pu ajouter leur grille d'occupation. L'interface de contrôle client en Qt a été réalisée par Sophie. Enfin, le simulateur de robot et d'environnement a été fait par Flavien.

2.3 Les problèmes et les solutions envisagées

Nous avons rencontré différents problèmes dès le début du projet.

SmarDTV souhaitait que notre code soit versionné mais que ce ne soit pas sur un système public ou hébergé par une société externe. Le git d'Epitech ne peut être diffusé que pour les personnes d'Epitech et il n'existe pas de comptes invités.

Le git privé sur une société externe n'a pas été retenu par SmarDTV pour des questions de sécurité et de confidentialité. La dernière solution consistait à héberger un git et à le configurer nous même au sein d'Epitech Marseille. La solution a été retenue mais n'a pas été mise en place car elle n'était pas prioritaire elle n'entravait pas le développement du projet, ni le suivi des avancements par SmarDTV.

Cela dit ce point reste à envisager dans le devenir du projet.

Lors de nos recherches concernant l'architecture nous avons abordé l'éventualité d'utiliser le module PIXMAP. Le module PIXMAP est à l'origine développé pour la Kinect et permet une modélisation 3D de l'environnement. Il est difficile d'adapter le module PIXMAP aux différents devices TANGO surtout à ce stade du développement du projet car la solution PIXMAP est plus technique. Cela reste tout de même une éventualité dans la prochaine évolution du projet.

Une fois le développement commencé, nous avons réalisé tous nos tests sur le simulateur, le temps d'avoir le robot Ubyx à disposition. Les résultats des tests étaient cohérents et les déplacements correspondaient aux commandes envoyées. Nous avons ensuite pris possession du robot et nous nous sommes aperçus que les résultats étaient complètement différents. Nous avons donc découvert la limite liée au robot qui est le manque de précision. En effet, en réalisant les trajets et leur retour, nous nous sommes aperçus que le déplacement du robot n'était pas précis et qu'il ne revenait pas exactement à sa position initiale.

Une nouvelle donnée est donc entrée en jeu : l'odométrie. L'odométrie est une technique qui permet de récupérer la position du robot et donc d'estimer le déplacement effectué. Nous avons donc récupéré

l'odométrie, mais nous nous sommes aperçus que chacune des données sur un même trajet était totalement aléatoire et ne correspondait à aucune distance.

Nous avons dû changer la carte du robot et il n'y avait pas d'odométrie. Après ce changement, nous avons pu nous concentrer sur la précision du déplacement du robot.

Ainsi grâce à ce programme nous avons pu réaliser des déplacements et des trajets avec leur retour, mais le résultat final manquait de précision, le robot ne revenait pas exactement à sa position initiale. Grâce à cette étape nous avons appris comment gérer une communication avec un robot, lui envoyer des requêtes qui lui, les interprètes et les traduits en mouvements.

Pour ce qui est de l'analyse de l'environnement, nous avons également rencontré des problèmes de précision quant aux données perçues par la tablette TANGO. Nous avions pour mission de créer une grille d'occupation qui aurait permis au robot de se déplacer de manière autonome en temps réel. Cette carte bidimensionnelle a été créée, seulement les données de distances envoyées par la tablette étaient difficilement interprétables car elles étaient envoyées en un temps trop long et en trop grand nombre. Les vues n'étaient pas affichées assez rapidement pour prétendre à un pilotage du robot en temps réel.

3. RESULTATS ET PERSPECTIVES

3.1 État final du projet

À ce jour, les différents points fonctionnant dans les applications sont les suivants :

Premièrement, l'ensemble du serveur permettant la liaison ROS avec les différents systèmes (l'application de la tablette Tango, le simulateur, l'application de contrôle client) sont opérationnels. Ce qui signifie que la première partie du cahier des charges est respectée.

Deuxièmement, l'application de pilotage développée sur la tablette Tango permet tantôt d'effectuer les actions reçues de l'utilisateur, tantôt de basculer sur un mode hybride "patrouille" permettant d'effectuer un retour sur ses pas de façon sommaire. Nous comptons en cours de développement, l'analyse de l'environnement, un mode "patrouille" autonome. Nous pouvons considérer avoir rempli approximativement 70% de ce point du cahier des charges.

Concernant le contrôle client, l'interface est simple et permet le déplacement du robot. Bien que dans un second temps, il faudrait développer un visuel pour la caméra du robot, nous pouvons estimer avoir rempli une attente supplémentaire.

Un simulateur permet de charger un environnement variable et simuler un robot retournant différentes mesures générées aléatoirement. Le simulateur répond aux attentes définies.

Les différents systèmes sont connectés par une architecture ROS qui est aussi un point essentiel attendu.

Globalement, nous avons atteint les objectifs revisités du cahier des charges. Concernant l'évolution du projet, de nombreuses améliorations et extensions sont possibles.

3.2 Améliorations et extensions possibles

Le projet LittleThumb a été réalisé dans une version très simplifiée, il reste néanmoins de nombreuses imperfections et des points à améliorer :

Sur le plan application, nous pouvons citer, l'analyse de l'environnement, le retour autonome à la base, les options du simulateur, le design des différentes interfaces visuelles (système de contrôle, simulateur).

Sur le plan matériel, idéalement, il faudrait corriger les données reçues des senseurs (ou changer le système qui les renvoie). Il faudrait envisager un autre device d'asservissement (pour la partie senseur).

CONCLUSION

Dans le cadre de notre projet, nous devions réaliser une application d'asservissement de robot avec un système de retour autonome sur la base de chargement. Dans un premier temps, nous avons dû nous documenter sur le système ROS ainsi que sur les technologies et les programmes existants dans le domaine de la robotique. Ensuite, nous avons effectué un travail sur la base fournie par SmarDTV, dans le but de comprendre le fonctionnement, la mise en œuvre de ROS. Nous avons ensuite reproduit le système mis en place par SmarDTV en dehors de la machine virtuelle, nous l'avons configuré et rendu prêt à être installé. Enfin nous avons pu commencer le développement du LittleThumb.

À travers ces différentes exigences imposées par le cahier des charges, nous aboutissons au résultat final qui est le suivant : L'intégralité des fonctionnalités basiques a été retranscrite concernant le serveur, le client de contrôle, le simulateur et le système d'asservissement. Au niveau de l'interface et de l'ergonomie, cette dernière est simplifiée par une présentation claire et concise. Côté développement, le projet a été pensé de façon à permettre à d'autres développeurs de reprendre et maintenir le code simplement.

Le projet LittleThumb, l'utilisation de la tablette Tango, ainsi que la découverte du fonctionnement et de l'utilisation du système ROS est une expérience bénéfique pour deux raisons :

D'une part, la nouveauté de la tablette Tango, la façon de l'utiliser et développer sur celle-ci, nous permet d'avoir une longueur d'avance sur le domaine, de maîtriser une technologie potentiellement recherchée au sein des entreprises. La technologie ROS quant à elle nous a permis de réellement nous plonger dans le monde de la robotique et de nous faire une expérience recherchée par beaucoup dans ce secteur.

D'un point de vue humain, le travail en partenariat avec des entreprises nous a permis de nous former sur de nouvelles exigences et contraintes, mais aussi de pouvoir réellement s'investir sur un projet concret, pensé et avec des moyens plus importants. La rencontre des différents acteurs du projet a été très enrichissante tant sur les contacts que sur les perspectives professionnelles.