Rapport de projet
Robot in the Loop 2

TW53 – **Projet Industriel**

Zhi Yan / Nathan Crombez / EricGete

utbm
université de technologie
Belfort-Montbéliard

EQUIPE

Majd **MASRI**

Irène **BON**

Axel **HEREN**

Département IMSI : Site de Belfort

utbm
université de technologie
Belfort-Montbéliard

Sommaire

1. Introduction

Les robots représentent une partie importante dans l'industrie 4.0 et sont les principaux acteurs de demain. Dans la logistique interne industrielle, un AGV (AutomatedGuidedVehicle) joue un rôle clé.

L'utilisation de plusieurs AGVs peut augmenter considérablement la productivité d'une industrie. Cependant, tester et déployer tous les vrais AGV est coûteux. Dans le cadre de ce projet, nous avons étudié un scénario où il est possible de mélanger des robots réels et des robots virtuels. Le but étant de permettre aux entreprises de demain d'explorer des solutions logistiques à faible coût et envisager l'ensemble des difficultés avant de déployer des robots en conditions réelles.

Notre projet repose entièrement sur le système d'exploitation robotique à la pointe de la technologie, i.e. ROS. Les travaux précédents (réalisés par les étudiants de la filière IMSI) ont montré des résultats prometteurs. Dans ce projet "Robot-in-the-loop 2", nous avons l'ambition d'aller encore plus loin en nous appuyant essentiellement sur la base de travail existante.

2. L'équipe

Notre équipe est composée de trois étudiants IMSI. Chacun d'entre nous a une fonction qui lui a été attitrée après capitalisation des compétences. Ce titre est une manière de fixer, dès le début de notre projet, les axes à suivre pour chacun.

Axel Heren : Chef de projet
Axel.heren@utbm.fr

Irène Bon : Correspondante communication & Rédactrice
Irene.bon@utbm.fr

MajdMasri : Gestionnaire du projet
Majd.masri@utbm.fr

Le nom de notre équipe est Slow Turtle Team, le terme turtle fait directement référence au Turtlebot utilisé pendant le projet. L'adjectif slow lie notre équipe à la tortue qui est réputée comme étant lente, nous avons relativement mis beaucoup de temps à comprendre l'ensemble du projet et comment se servir de tout les outils de ce nouvel OS.

Ce rapport vient en complément des 2 premiers rapports associés à ce sujet.

3. Le projet

3.1. L'état des lieux du projet précédent A2018

Nous avons commencé en réalisant un état des lieux du projet de ce qui a été réalisé par l'équipe précédente pendant le semestre d'automne 2018. Leur objectif était de contrôler un robot virtuel et réel et de créer un espace collaboratif où les deux robots évolueraient.

Ils n'ont malheureusement pas atteint leurs objectifs à cause d'un problème de synchronisation entre les deux types de robots.

3.2. L'objectif du projet

L'objectif principal du projet est donc de réussir la synchronisation entre le robot réel et le robot virtuel dans un environnement réel modélisé.

Nous avons séparé le projet en trois grandes phases : nous avons d'abord commencé par une prise en main complète du projet. Cette étape passe par l'apprentissage de l'utilisation des différents logiciels utilisés et le fonctionnement global du Turtlebot. Nous nous sommes également appropriés les expériences et connaissances des groupes précédents. Par la suite, nous avons réalisé des essais de commande de robot, de reconnaissance de l'environnement et nous avons appréhendé quelques contraintes que nous n'avions pas prévues dans le projet. Pour finir, nous avons apporté les modifications nécessaires aux objectifs finaux et nous avons finalisé la création d'un environnement physique dans lequel évolue le robot.
Nous avons également accordé de l'importance sur l'avenir envisageable et souhaitable pour le projet.

Comment vous pouvez le constater dans la description de nos objectifs, nous n'avons pas pu atteindre l'objectif principal, soit la synchronisation entre le robot réel et virtuel. Les obstacles nous empêchant d'atteindre cet objectif seront détaillés dans la suite de ce rapport.

3.3. Les livrables

Le client nous a fait part de ses attentes au niveau des livrables lors de notre première réunion.

Pour chacune des trois phases : un diaporama de présentation et une soutenance orale validée par un compte-rendu de revue de projet signé par le responsable du projet.

Pour la soutenance finale : Un rapport de projet résumant le travail effectué ainsi qu'une démonstration du robot en parallèle de la présentation orale.

3.4. Besoins et contraintes

Ce projet est naturellement soumis à des contraintes induites par l'environnement dans lequel il évolue mais aussi des contraintes d'ordre temporelle et des compétences limitées que nous avons dans le domaine de la robotique.

C'est pour cette raison qu'une grande première phase d'apprentissage nous était indispensable pour apprendre et comprendre les bases de la programmation sous Linux. Le fonctionnement et les caractéristiques du robot ainsi que ses composants furent également étudiés.

Le temps imparti pour ce projet étant limité à 14 semaines, nous avons dû nous organiser pour pouvoir travailler sur le robot malgré les contraintes de disponibilité entre nous, entre notre groupe et les professeurs qui nous aident dans ce projet et les contraintes d'horaires d'ouverture des salles où est stocké le matériel (robot, ordinateur portable, etc…).

3.5. Ressources

Les ressources utilisées pour mener à bien ce projet sont les suivantes :

- Plusieurs ordinateurs dont un sous Linux
- Un routeur pour créer un réseau local
- Softwares : Gazebo, ROS et RViz
- Un robot Turtlebot Burger avec ses différents composants :
    - Ordinateur de bord unique Raspberry Pi 3.
    - Capteurs : Odometry, LIDAR
    - Une structure évolutive
    - Open CR 32bit ARM Cortex-M7
    - 2x Dynamixel pour les roues
    - 2x roues dentées pour pneumatiques et chenilles
    - Batterie Li-Po 11.1V 1.8000mAh
- Divers équipements disponibles dans la salle de travail (salle B412 du bâtiment B sur le site de Belfort)

Remarque : Certaines ressources ne nous ont pas été fournies dès le début du projet, comme par exemple un routeur qui nous était essentiel pour effectuer la communication entre le robot et l'ordinateur. Ce manque de matériel nous a freiné dans notre projet.

3.6. La planification du projet

Le diagramme de Gantt est un outil qui représente visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches (générales) à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet, dans notre cas en semaines.

Voici le Gantt du projet Robot in the Loop – **Version 1.1** :

Lors de la structuration initiale (version 1.1) du notre projet nous avons réfléchi afin de découper le projet en plusieurs parties, en plusieurs jalons, cruciaux à son bon déroulement sur le long terme. Cette décomposition des tâches pourrait se résumer grossièrement en 3 grandes phases

1) PRISE EN MAIN DU PROJET : Etude des précédents travaux, du robot, de ses composants et des logiciels
2) REALISATION DES SIMULATIONS : Recherche sources du problème de synchronisation, recherche de solutions et essais
3) AMELIORATION CONTINUE : Suite des recherches et/ou amélioration

De plus, nous nous sommes basés sur la disponibilité que nous avions en début de projet, en nombre de jours. Ensuite et grâce aux méthodes de planification de projet apprises dans notre cursus nous avons réparti la durée totale parmi les tâches du projet en priorisant les activités les plus importantes.
Un exemple de tâche considérée importante est, par exemple, la compréhension du travail réalisé par l'équipe précédente ou alors l'étude du robot et de ses composants.

Certains de ces points n'ont pas ou pas totalement été développés à cause d'évènements inattendus ce qui a perturbé notre planning de départ.
Voici la version 1.2 de notre Gantt, on peut y observer des tâches qui ont été ajoutées ce qui a entrainé un retard des tâches initialement planifiées.

Diagramme de Gantt - Version 1.2

Groupe TW53 - Projet Robots in the loops 2 - Irène Bon - Majd Masri - Axel Heren

Date de début et fin du projet : 26/09/2019 - 16/01/2020

Sujet supervisé par Dr Zhi Yan

REVUE 1 | REVUE 2 | REVUE 3

	Mois	Septembre		Octobre				Novembre				Décembre				Janvier			Début (Semaine)	Fin (Semaine)	Durée (Semaines)
	Semaines	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3			
PHASE 1 : PRISE EN MAIN DU PROJET																					
DERSTAND (étude de l'exist	Etat des lieux des projets précédents																				
	Etude des softwares et OS (Linux)																				
	Etude du robot et ses composants																				
	Prendre connaissance des programmes existants																		39	43	4
	Simulation de l'état actuel																				
	Rédaction des documents d'organisation																				
	Rédaction des différentes fiches synthèses																				
PHASE 2 : REALISATION DES SIMULATIONS																					
	Résolution Problème de connexion																				
	Résolution Problème moteurs roues																				
	Résolution Problème commande robot																				
SIMPLIFY (Trouver et préparer des solutions)	Recherche de solutions																				
	Proposition de solutions																				
	Validation des solutions envisageables																		45	49	0
	Révision de la simulation																				
	Réalisation des tests de simulation																				
	Analyse des résultats obtenus																				
	Recherche de points d'améliorations																				
	Bilan de l'expérimentation																				
PHASE 3: AMELIORATION CONTINUE																					
	Modification de la situation initiale																				
	Réalisation des tests de simulation																				
	Validation de la solution finale																				
ACT (Mise en œuvre)	Estimation du gain apporté																			3	4
	Bilan du projet dans son ensemble																				
	Perspectives à envisager pour l'avenir																				
	Rédaction des différents livrables																				
LIVRABLES UTBM																					
	Cahier des charges du projet																				
	Livrables revue de projet 1																				
CONDITIONS UV TW53	Livrables revue de projet 2																		39	3	12
	Livrables revue de projet 3																				
	Rapport final et présentation finale																				
	Fiche finale de synthèse du projet																				

Pour donner suite à la mise à jour de nos objectifs après la phase 2, voici le planning réel que nous avons suivi pour mener à bien notre projet. En rouge sont indiquées les taches initialement prévues qui n'ont pas été réalisées. On peut observer que de nouvelles tâches ont été ajoutées en phase 3.

Diagramme de Gantt - Version 1.3

Groupe TW53 - Projet Robots in the loops 2 - Irène Bon - Majd Masri - Axel Heren

Date de début et fin du projet : 26/09/2019 - 17/01/2020

Sujet supervisé par Dr Zhi Yan

REVUE 1 | REVUE 2 | REVUE 3

	Mois	Septembre		Octobre				Novembre				Décembre				Janvier			Début (Semaine)	Fin (Semaine)	Durée (Semaines)
	Semaines	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3			
PHASE 1 : PRISE EN MAIN DU PROJET																					
DERSTAND (étude de l'exist	Etat des lieux des projets précédents																				
	Etude des softwares et OS (Linux)																				
	Etude du robot et ses composants																				
	Prendre connaissance des programmes existants																		39	43	4
	Simulation de l'état actuel																				
	Rédaction des documents d'organisation																				
	Rédaction des différentes fiches synthèses																				
PHASE 2 : REALISATION DES SIMULATIONS																					
	Résolution Problème de connexion																				
	Résolution Problème moteurs roues																				
	Résolution Problème commande robot																				
	Résolution Problème reconnaissance environnement																				
SIMPLIFY (Trouver et préparer des solutions)	Recherche de solutions																				
	Proposition de solutions																				
	Validation des solutions envisageables																		45	51	7
	Révision de la simulation																				
	Réalisation des tests de simulation																				
	Analyse des résultats obtenus																				
	Recherche de points d'améliorations																				
	Bilan de l'expérimentation																				
PHASE 3: AMELIORATION CONTINUE																					
	Création d'un environnement physique																				
	Reconnaissance de l'environnement																				
	Visualisation du robot sous Gazebo																				
	Médisation de la carte finale																				
	Essais commande robot																				
ACT (Mise en œuvre)	Modification de la situation initiale																		2	3	2
	Réalisation des tests de simulation																				
	Validation de la solution finale																				
	Estimation du gain apporté																				
	Bilan du projet dans son ensemble																				
	Perspectives à envisager pour l'avenir																				
	Rédaction des différents livrables																				
LIVRABLES UTBM																					
	Cahier des charges du projet																				
	Livrables revue de projet 1																				
CONDITIONS UV TW53	Livrables revue de projet 2																		39	3	14
	Livrables revue de projet 3																				
	Rapport final et présentation finale																				

4. Corps du projet

4.1. Phase 1 : Prise en main du projet

Une fois le planning créé et le cahier des charges rédigé, nous nous sommes lancés dans la première phase du projet : L'étude du projet.

4.1.1. Etude des travaux précédents

Le premier rapport, émis par Messieurs Bellini, Cedomirovic et Delorme, dans le cadre de l'UV TW53 au semestre de printemps 2018, consistait en l'assemblage et le paramétrage d'un TurtlebotWaffle (notre modèle à nous est le Burger) afin de réaliser une démonstration vidéo de ses fonctionnalités dans un cadre logistique.

Le deuxième rapport, émis par Messieurs D'ambelle et Pourzergues et Madame Li Crapi, dans le cadre de l'UV TW53 au semestre d'automne 2018, consistait à coordonner plusieurs robots (réels et virtuels) sur le même logiciel ROS afin d'optimiser un espace de travail collaboratif de robots mobiles industriels.

4.1.2. Etude des softwares et OS (Linux)

L'ensemble du projet nécessite l'utilisation de l'OS Linux car c'est le seul OS supportant notre partie Software **Ubuntu** est un système d'exploitation Linux basé sur la distribution Linux Debian.

Notre projet se déroule sous le logiciel de contrôle **ROS** (Robot Operating System). C'est un middleware qui a été conçu pour programmer les robots. ROS fournit une importante collection d'outils et de librairies.

On y retrouve les **paquets**, ce sont les différents besoins du système. Chaque paquet exprime un besoin. Par exemple, s'il faut calculer la trajectoire d'un robot, un paquet sera créé pour ce besoin.

Les **nœuds** sont des processus qui permettent d'effectuer un calcul. Chaque nœud a un but. Par exemple, il y a un nœud qui permet de déplacer un robot jusqu'à un point, un deuxième nœud permettant l'arrêt du robot.

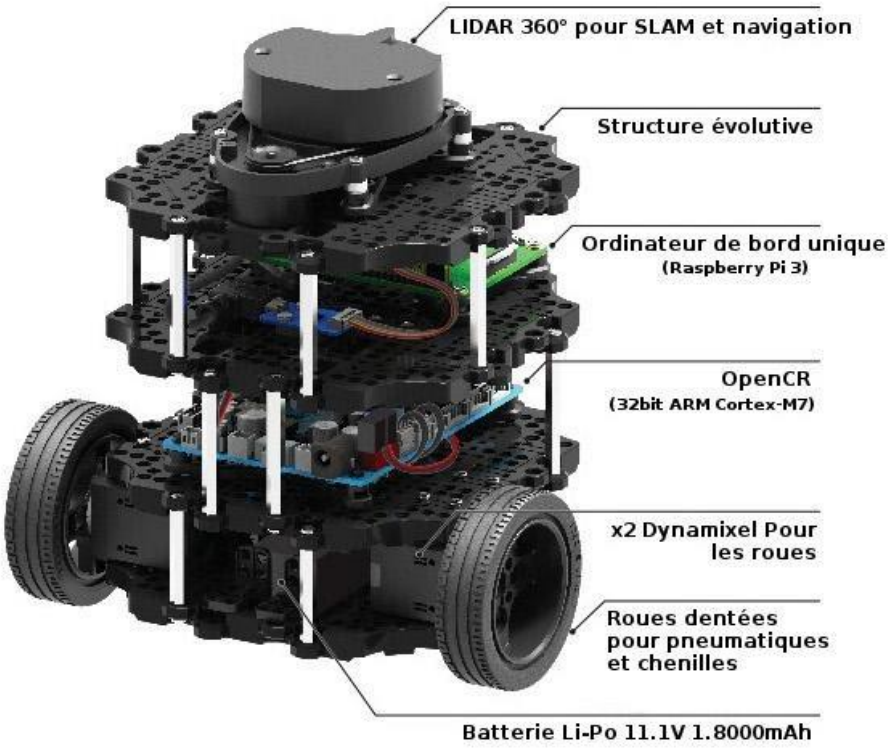
Le **Master** de ROS permet aux nœuds de se reconnaitre et de communiquer entre eux.

Gazebo : Ce Software permet de simuler de manière dynamique et avec une vue en 3D les déplacements des robots.

RViz : logiciel de visualisation en complément de Rviz mais avec la possibilité d'ajouter un environnement et des objets virtuels.

Dans la globalité du projet, la partie Software est open source et notre groupe n'était initialement pas formé à l'utilisation de ces Softwares, il a donc fallu apprendre les bases pour commencer à avancer.

4.1.3. Etude du robot et de ses composants



Voici quelques caractéristiques du Turtlebot 3 Burger :

Caractéristiques	Burger
Vitesse translationnelle maximale	0,22 m/s
Vitesse rotationnelle maximale	2,84 rad/s
Charge maximale	15 kg
Dimensions (L x L x H) en mm	138 x 178 x 192
Poids	1 kg
Temps de fonctionnement	2,5 h
Temps de recharge	2,5 h
Contrôleur	Raspberry Pi 3
Actionneurs Dynamixel	XL430-W250-T
Capteurs	LiDAR 360° Gyroscope 3-Axes Accelerometer 3-Axes Magnetomete 3-Axes

Le LIDAR 360° : Ce capteur utilise la lumière pour déterminer la géométrie de l'espace autour du robot. En tournant sur lui-même il peut ainsi détecter l'ensemble de l'environnement et sa distance à 360 degrés.

Le robot utilise aussi l'**odométrie** pour déterminer sa position dans l'espace en fonction du nombre de tours effectués par les roues ce qui définit la distance parcourue par rapport à une position de base.

Le **Raspberry Pi** est un nano-ordinateur monocarte à processeur ARM (Architecture externe). Il permet les échanges d'informations entre l'ordinateur et le robot.

L'Open CR signifie Open-source Control module for ROS. C'est la carte de contrôle du robot, elle permet de connecter tous les éléments du robot entre eux.

4.1.4. Situation initiale : début des premiers essais

Suite à nos recherches sur les points précédents, nous étions impatients de commander le robot. Malheureusement nous avons rencontré un problème de connexion entre le robot et l'ordinateur que nous essayions de connecter en Wi-Fi.
En effet le robot et l'ordinateur communiquent par Wi-Fi. Nous devions avoir un accès déporté par rapport au robot.Nous devions nous connecter à un réseau local et il nous fallait une adresse IP. L'accès au réseau est nécessaire car nous voulions la conserver lorsque nous nous connections au réseau.
Nous avons donc rencontré un problème de connexion au robot réel et configuration de l'adresse IP.

- Le robot reconnaissait le réseau local qu'on a créé, mais il refusait de s'y connecter.
- Nous avons alors testé sur un autre réseau ouvert. Tout était connecté, mais la commande ssh dans le Shell ne répondait pas.

Après plusieurs jours d'essais infructueux pour trouver une solution en essayant de reproduire le travail des groupes précédents, Monsieur Sihao Deng, enseignant chercheur à l'UTBM nous aidé à régler le problème en fournissant un routeur. Le robot et l'ordinateur sout tous les deux connectés au routeur, ils peuvent donc communiquer entre eux.

La source du problème est que nous ne savions pas qu'un routeur était indispensable pour la connexion du robot à l'ordinateur, ça n'était pas un élément que nous aurions pu trouver dans les rapports précédents et nos faibles connaissances dans ce domaine ne nous ont pas aidés non plus.

4.1.5. Bilan de la phase 1

La phase s'est très bien déroulée, nous avons bien compris les travaux effectués précédemment, nous avons commencé notre apprentissage sur les softwares et nous nous sommes renseignés sur le fonctionnement du robot et ses composants. Nous avons respecté le planning et suite à notre première revue de projet, monsieur Zhi Yan, responsable de notre projet, a validé notre travail.

4.2. Phase 2 : Réalisation des simulations

Ce que nous ne savions pas c'est que ce problème de connexion rencontré dans la phase 1 nous prendrait autant de temps. En addition avec d'autres évènements imprévisibles, nous avons pris du retard sur la phase 2 qui était initialement dédiée à la réalisation des simulations afin de synchroniser le robot réel au robot virtuel.

4.2.1. Problèmes rencontrés

Roues en dysfonctionnement

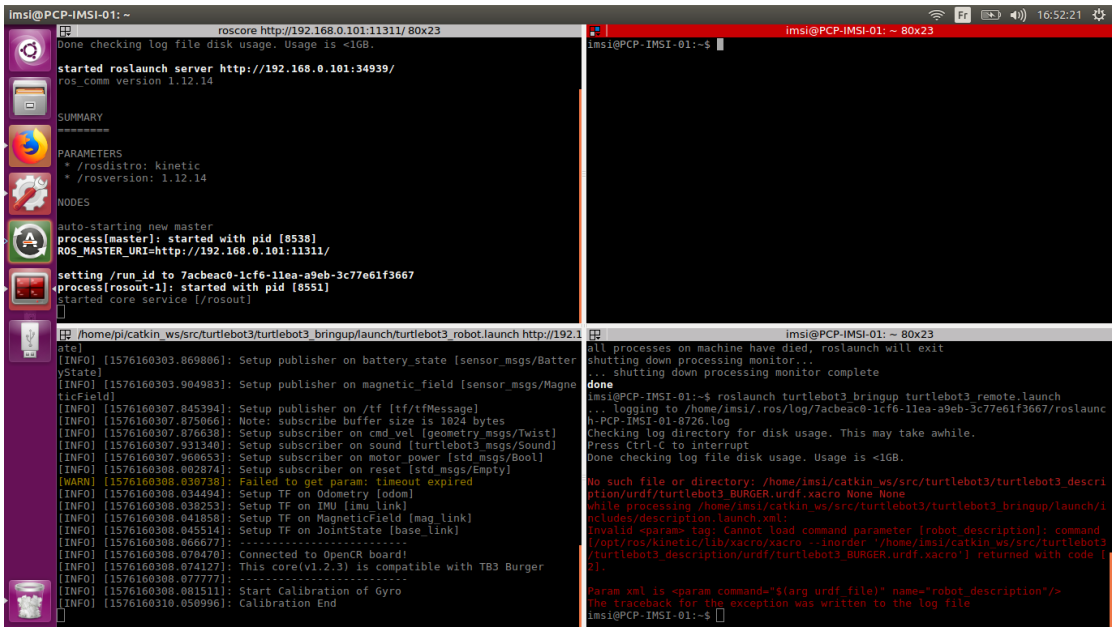
Une fois le problème de connexion résolu, nous avons rencontré une panne des moteurs faisant tourner les roues.
Nous n'arrivions pas à comprendre pourquoi et nous avions d'abord pensé à un problème mécanique qui n'était pas de notre ressort.

Monsieur Zhi Yan a donc cherché la cause du problème et nous l'a décrite :

Le robot étant stocké dans une salle commune et plusieurs personnes travaillant sur le robot, il est possible que quelqu'un ait, par inadvertance, modifié un des programmes concernant la commande des roues. Etant donné qu'il n'était pas possible de retrouver la modification dans le programme, monsieur Zhi Yan a désinstallé puis réinstallé le programme pour l'initialiser.
Les moteurs fonctionnent donc comme avant.

Reconnaissance de l'environnement

Le robot pouvait maintenant se déplacer. Cependant il ne reconnaissait pas l'environnement autour de lui. La fonction qui est faite pour, ne trouve pas l'emplacement du logiciel de visualisation RViz qui nous permet d'accéder aux "Interactive Marker" alors que, quand on regardait dans l'emplacement indiqué, on pouvait constater que la fonction cherchée était bien à cet emplacement.



Ce problème a été résolu par Monsieur Zhi Yan.

Plan environnement robot

L'équipe qui avait travaillé précédemment sur les environnements (virtuel et réel) du robot ne nous a pas laissé les plans pour réaliser l'environnement réel. Nous devions donc le recréer.

Proposition de solution :

Sur plusieurs feuilles A1 attachées entre elles, nous allons tracer les emplacements des murs et des obstacles présents dans l'environnement.

Nous modéliserons ensuite ces éléments (ex: cartons) que nous pourrons placer sur le plan en feuille A1.

Le robot réel pourra se déplacer dans un environnement identique à celui du robot virtuel.

Cela permettra aux autres groupes qui travailleront sur ce projet de reprendre cet environnement.

4.2.2. Retard dans le planning

Suite à ces évènements imprévisibles, nous ne pouvions pas travailler sur la synchronisation du robot comme il était initialement prévu dans la phase 2, l'atteinte de notre objectif principal avant la fin du semestre nous semblait donc impossible.

4.2.3. Mise à jour des objectifs

Suite à la deuxième revue de projet avec monsieur Zhi Yan, nous avons décidé de mettre à jour nos objectifs de manière à les rendre plus accessibles.

Objectifs initiaux : la synchronisation du robot réel et virtuel dans un environnement identique.

Nouveaux objectifs : Pouvoir rendre compréhensible le travail effectué aux prochains groupes, en expliquant les problèmes rencontrés mais surtout en expliquant pourquoi ça n'a pas fonctionné et en donnant si possible une solution.

L'important dans ce projet qui traite des domaines de l'informatique et de la robotique, des domaines qui nous sont peu connus, c'est principalement d'apprendre et de comprendre, l'atteinte des objectifs n'est pas essentielle.

4.2.4. Bilan de la phase 2

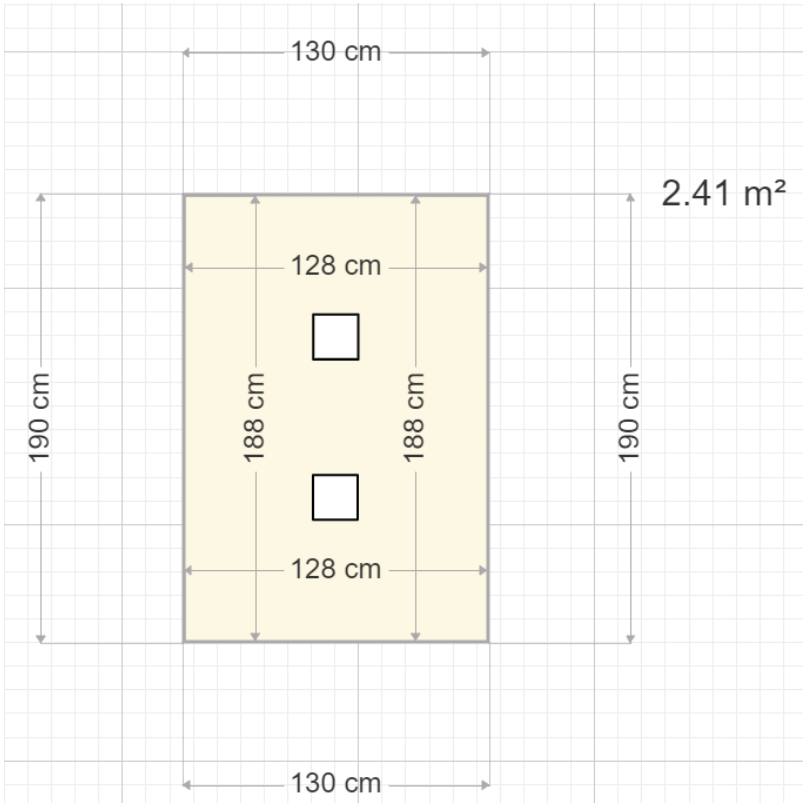
Suite à la mise à jour de nos objectifs nous avons décidé de nous concentrer sur la reconnaissance de l'environnement, pour pouvoir laisser aux prochains groupes une base solide et net du travail déjà réalisé qu'ils pourront utiliser pour travailler sur la synchronisation des robots réels et virtuels.

4.3. Phase 3 : Avancement et base de travail pour les prochains groupes

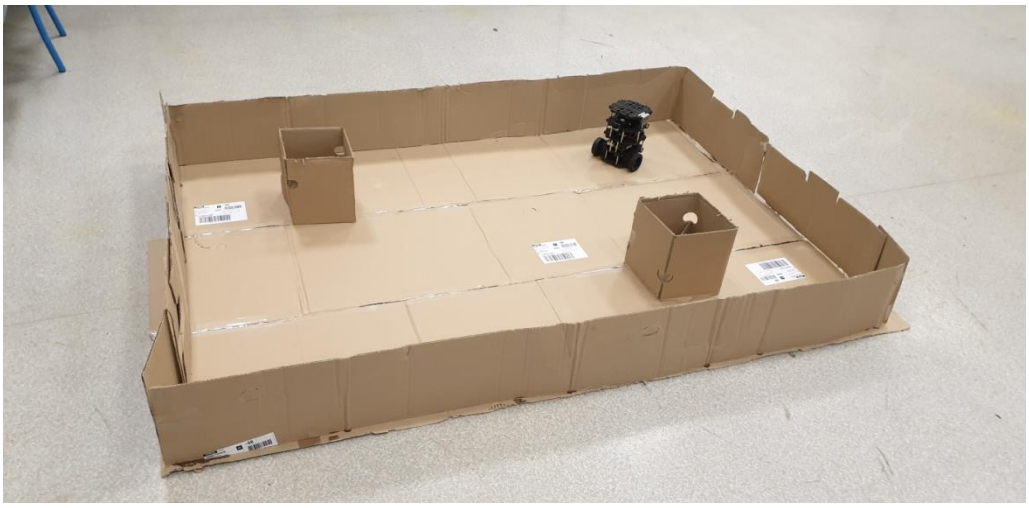
Cette phase 3 est donc dédiée à la création et la reconnaissance de l'environnement, ainsi qu'aux commandes du robot utilisées pour évoluer dans cet environnement. A partir de ce travail, les prochains groupes pourront avancer, nous l'espérons, plus facilement et rapidement.

4.3.1. Création d'un environnement physique

Nous avons créé un plan de l'environnement physique dans lequel va évoluer le robot réel. Sur la photo ci-dessous vous pouvez observer les dimensions choisies.



Par manque de temps et de budget, pour créer l'environnement réel nous avons utilisé des cartons que nous avons fixé ensemble à l'aide de scotch et de colle. Nous avons aussi créé des obstacles dans l'environnement qui peuvent être disposés où l'on veut dans celui-ci.

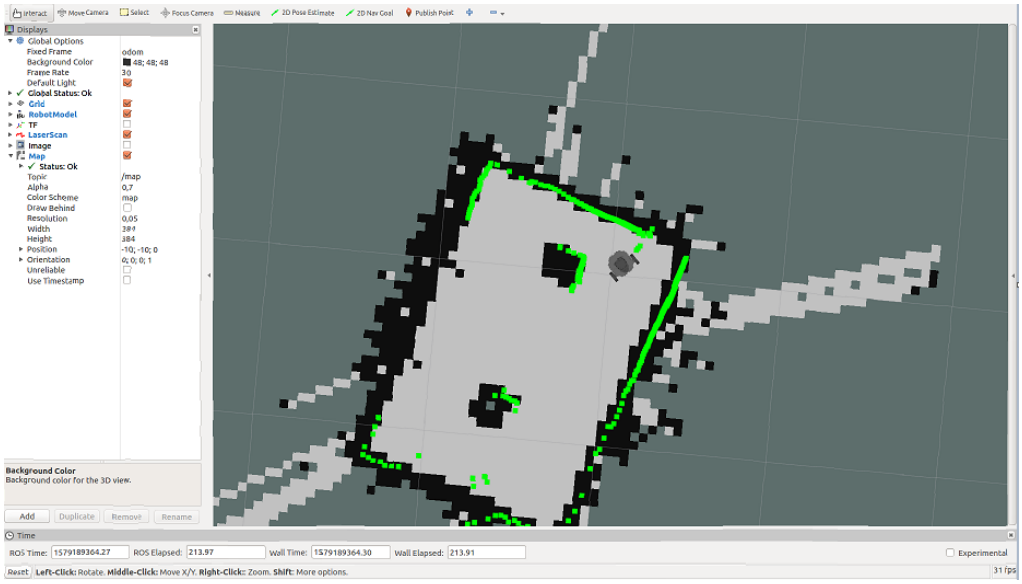


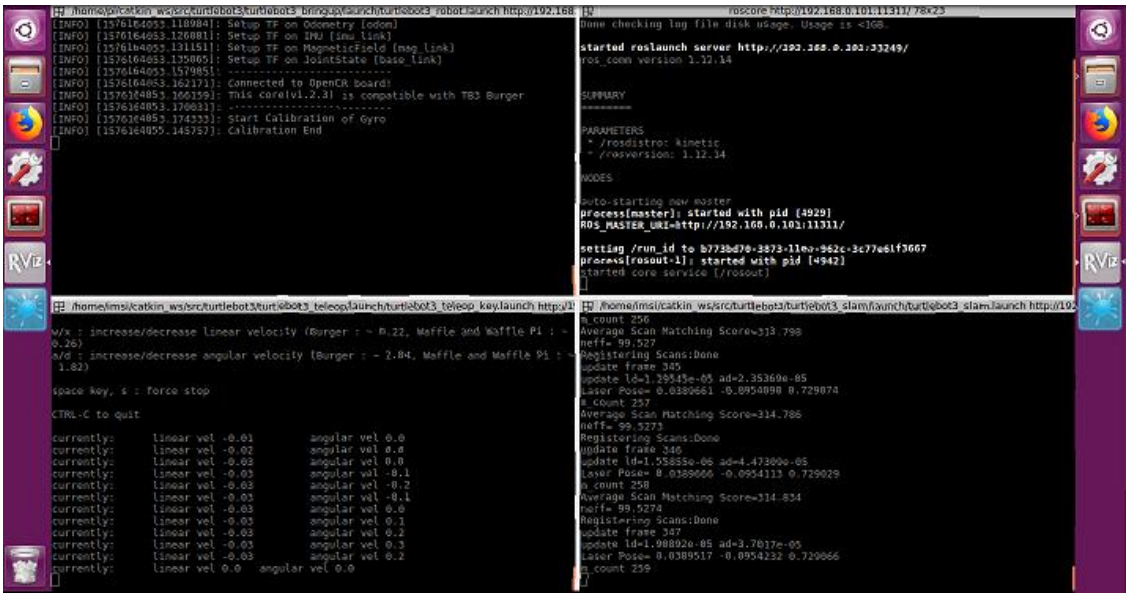
Nous considérons cet environnement réel comme la première version.
Les prochains groupes pourront reprendre cette configuration et recréer cet environnement avec des matériaux plus solides, peut être avec un meilleur agencement de manière à ce que l'environnement puisse être facilement portable et/ou stockable.

4.3.2. Reconnaissance de l'environnement

Afin que le robot réel analyse et reconnaisse l'environnement autour de lui grâce à ses capteurs nous avons effectué un **SLAM**. Ce sigle signifie en anglais Simultaneous Localization And Mapping. C'est un package qui permet au robot de construire la carte de son environnement et de s'y localiser. Plus précisément nous avons utilisé dans ce package l'algorithme **Gmapping**. La navigation ne peut être faite sans SLAM. Ils sont dépendants l'un de l'autre.

Grace à ce SLAM et au plan que nous avons créé, nous avons pu recréer l'environnement en virtuel en utilisant le programme building editor sous Gazebo.

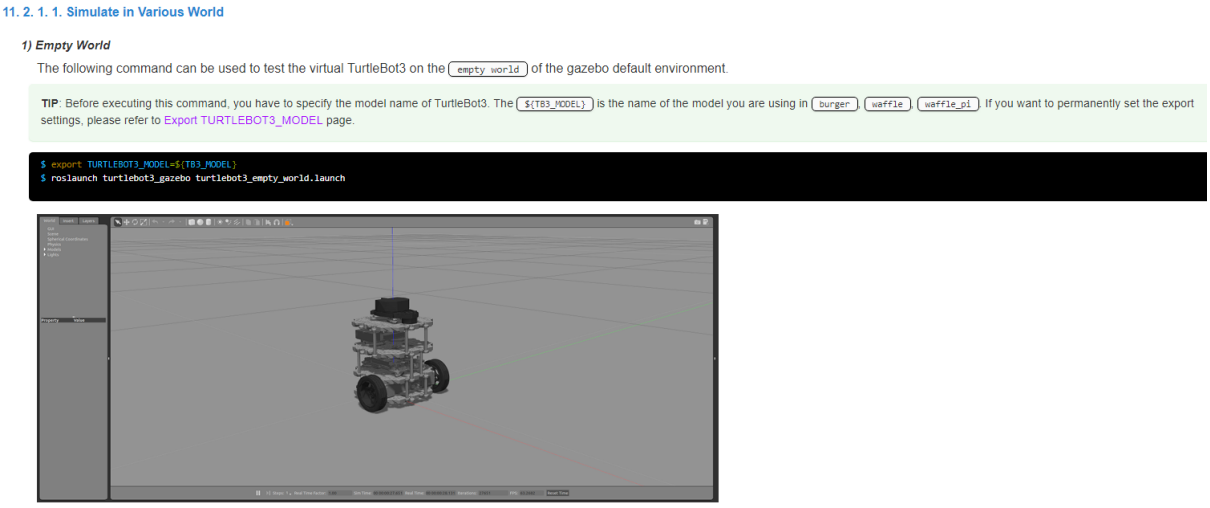




4.3.3. Visualisation du robot sous gazebo

Pour visualiser le robot sous Gazebo nous avons utilisé un tutoriel disponible sur ce lien :

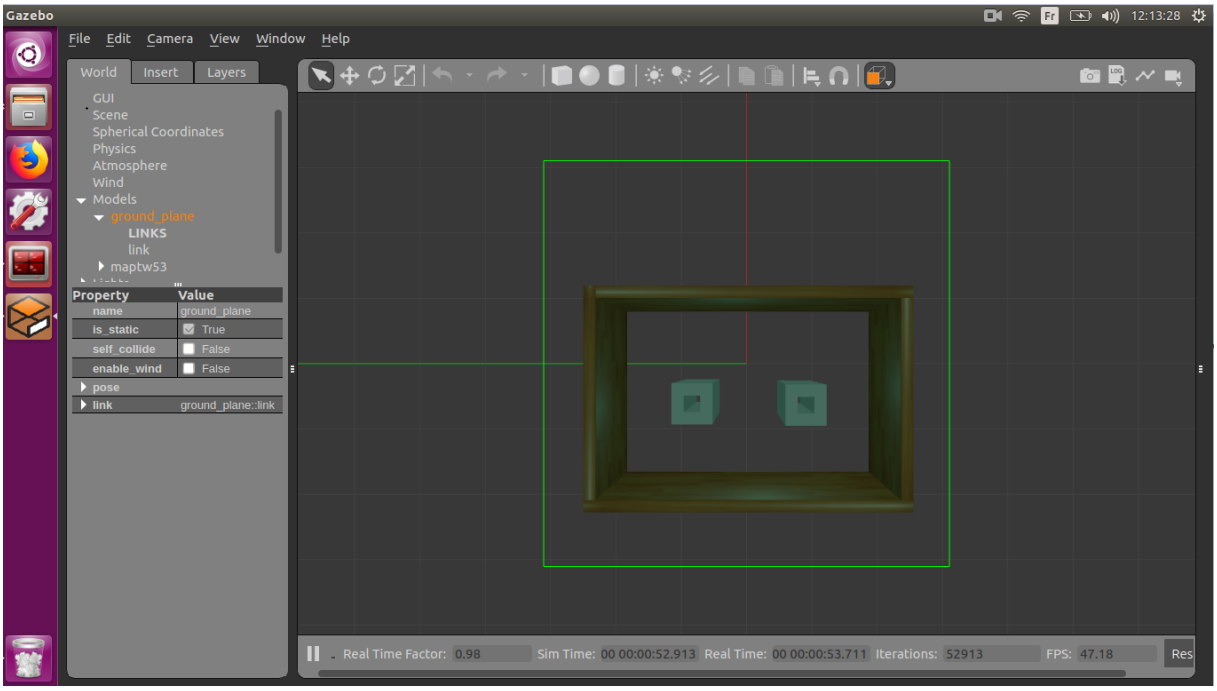
http://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#simulation



4.3.4. Modélisation de la carte finale

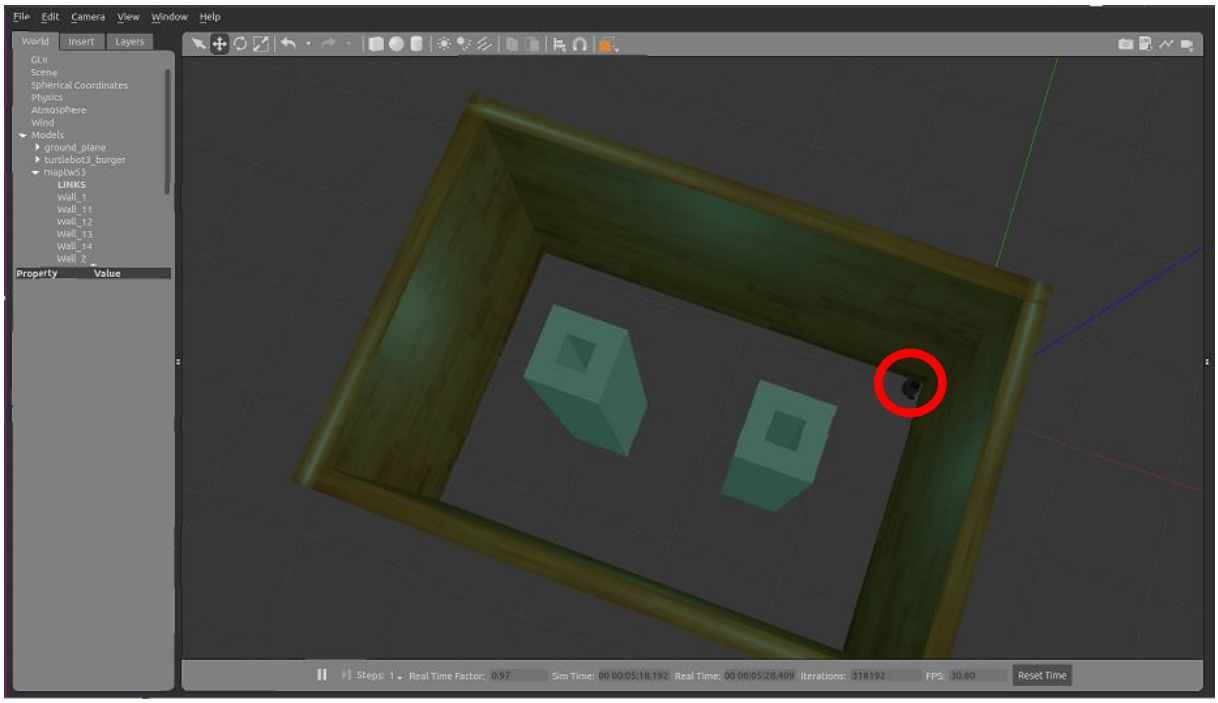
Sous Gazebo, nous avons pu modéliser la carte finale grâce au plan que nous avions créé.
A partir de building editor, on crée un fichier MAP qui contient un fichier model.config et un fichier model.sdf. Et c'est à partir de ce dernier fichier qu'on obtient un fichier model.world.

On obtient grâce à ce fichier model.world notre carte virtuelle.



4.3.5. Essai de synchronisation du robot

Nous avons pu par la suite appeler le robot dans cet environnement et nous l'avons placé dans un coin de la carte. Nous avons placé le robot réel sur la même position dans l'environnement réel et nous avons observé les mouvements des deux robots avec une commande manuelle.



4.3.4. Bilan de la phase 3

Par manque de temps nous n'avons pas pu pousser notre travail plus loin mais nous conseillons au groupe qui reprendra notre travail de récupérer la position du robot dans le référentiel du world grâce a un package créé par Monsieur Zhi Yan.
https://github.com/yzrobot/pose_publisher

Ce package donne la position et l'orientation du robot dans le référentiel de la carte.
Il faudra ensuite lancer le programme et faire un écho du topic sur lequel le programme publie.
Ce programme fonctionne sur le principe de transformées, les TF.
Il faudra également régler les horloges des deux robots pour permettre la synchronisation.

Nous vous invitons à regarder les différents fichiers présentés dans le dossier, vous trouverez les différents programmes utiles au fonctionnement du projet ainsi que notre état d'avancement sur la présentation PowerPoint.

Le robot réel envoie des informations en temps réel (positions et orientations) au programme. Ces mêmes informations sont disponibles pour le robot virtuel sous gazebo.

La structure du programme final est un fichier .launch ou l'on retrouve:

- l'environnement virtuel conforme à l'environnement réel **(fichier .world)**
- Le robot réel en action via un programme de trajectoire **(node 1)**
- Le robot virtuel en action via un autre programme de trajectoire **(node 2)**
- Une réaction anticollision lorsque les robots sont trop proches **(fichier .py)**

Les trajectoires s'exécutent via des programmes python **(fichier .py)**

5. Conclusion

Bien que nous n'ayons que très peu de connaissances dans le domaine de l'informatique et de la robotique, ce projet nous a permis de sortir de notre zone de confort et de nous confronter a un domaine qui nous semblait très abstrait. Nous avons pu enrichir nos compétences et notre culture informatique.
Nous avons également eu l'occasion de mettre en pratique des compétences que nous avions déjà acquises dans notre cursus comme la gestion de projet, le suivi des délais ou la résolution de problème.
Nous avons appris que quel que soit le projet sur lequel nous travaillons, il est important de revoir nos objectifs, de les changer si besoin en fonction de nos capacités et en accord avec le client.

D'un point de vue humain, ce projet fût l'occasion pour nous de vivre à nouveau l'expérience du travail en équipe. Nous avons chacun apporté nos compétences à ce projet et nous avons pu apprendre ensemble et nous soutenir quand parfois la motivation faiblissait un peu.

En somme, ce projet a été pour nous une expérience très enrichissante qui s'inscrivait parfaitement dans l'industrie 4.0.

Nous tenons à remercier Monsieur Zhi Yan, responsable et client de ce projet qui nous a soutenus et aidés tout au long de ce semestre.
Nous remercions Monsieur Sihao Deng, enseignant chercheur à l'UTBM qui nous a aussi aidés à régler quelques problèmes techniques auxquels nous étions confrontés parfois.
Nous remercions également Monsieur Eric Gete, Directeur de la filière IMSI et responsable de l'UV TW53 qui nous a donné beaucoup de conseils pour nos futures vies professionnelles lors de notre revue de projet finale à laquelle il a assisté.
Nous remercions enfin l'UTBM qui nous a permis de mener à bien ce projet grâce à la formation que nous avons reçue tout au long de notre cursus.

6. Recommandations

Voici quelques recommandations que nous conseillons aux futurs enseignants qui porteront ce projet ainsi qu'un groupe qui reprendra notre travail.

Nous n'avions aucunes formations informatiques avant de commencer le projet, nos compétences n'étant pas égales, une petite formation dans le format d'un cours de deux heures par exemple pour nous enseigner les premières bases nous aurait aidés à nous lancer plus facilement et plus sereinement dans le projet.

Bien que Monsieur Zhi Yan, ait participé le plus souvent possible à l'avancement de notre projet, ses contraintes de disponibilité nous ont freinés lors de situations de blocage. D'autant que Monsieur Nathan Crombez, Co-responsable du projet, était indisponible.

Nous recommandons aux futurs étudiants qui travailleront sur ce projet de se concentrer sur l'apprentissage et la transmission des connaissances acquises dans ce projet et de mettre à jour à plusieurs reprises si besoin les objectifs du projet, en accord avec leurs compétences.

7. Bibliographie

http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/

http://emanual.robotis.com/docs/en/platform/turtlebot3/export_turtlebot3_model/

http://gazebosim.org/tutorials?tut=model_editor

http://gazebosim.org/tutorials?tut=build_world#LoadingaWorld

http://www.kermani.us/index.php/tutorials/ros-tutorial/32-02-ros-tutorial-getting-started

https://wiki.ros.org/ROS/NetworkSetup

http://obconnect.com:8069/slides/slide/turtlebot3-37-gazebo-simulation-tutorial-2499

https://automaticaddison.com/how-to-launch-the-turtlebot3-simulation-with-ros/

http://mobotica.blogspot.com/2011/09/using-gazebo.html

http://gazebosim.org/tutorials?tut=ros_roslaunch

http://gazebosim.org/tutorials?tut=build_world&cat=build_world

http://answers.gazebosim.org/question/15245/sdf-and-world-file-correct-usage/

http://emanual.robotis.com/docs/en/platform/turtlebot3/applications/#automatic-parking

http://emanual.robotis.com/docs/en/platform/turtlebot3/bringup/#run-roscore

https://answers.ros.org/question/260930/what-do-three-framesbaselinkodommap-mean/