

Développement et intégration de ROS pour le robot KUKA Youbot



B. Mollard
basile.mollard@ensta-bretagne.org

Résumé

Dans le cadre de ma deuxième année d'école d'ingénieur en robotique à l'ENSTA Bretagne, j'ai effectué un stage de trois mois à l'université Van Amsterdam. Mon travail portait sur le développement de l'autonomie du robot *Kuka Youbot*, en partenariat avec Juliette Faury. L'objectif principal était de permettre au robot de se déplacer de manière autonome en utilisant un *Lidar Velodyne VLP16* couplé à une caméra, dans le but de réaliser du *SLAM* (*Simultaneous Localization and Mapping*). Nous nous sommes basés sur des travaux antérieurs qui utilisaient uniquement une caméra pour le SLAM, mais notre approche consistait à intégrer le Lidar pour améliorer la précision.

Dans cette collaboration, je me suis chargé de la calibration du Lidar et de la création d'un *node ROS* pour l'acquisition et le traitement des données. Pendant ce temps, Juliette travaillait sur l'intégration de la caméra et la configuration du robot. Cependant, en raison d'une erreur survenue après un mois de travail, nous n'avons pas pu atteindre nos objectifs finaux malgré nos efforts pour corriger le problème au cours des trois mois de stage.

Abstract

As part of my second year of engineering studies in robotics at ENSTA Bretagne, I completed a three-month internship at the University of Amsterdam. The project aimed to develop autonomous navigation for the Kuka Youbot robot, in collaboration with Juliette Faury. Our main objective was to enable the robot to navigate autonomously using a Velodyne VLP16 Lidar combined with a camera, applying Simultaneous Localization and Mapping (SLAM) techniques. This work built on previous research that used only a camera for SLAM, but our approach was to integrate Lidar data to enhance accuracy.

I was responsible for calibrating the Lidar and developing a ROS node to acquire and process the data, while Juliette focused on integrating the camera and setting up the robot. Unfortunately, due to an error that occurred after the first month of work, we were unable to fully achieve our initial objectives, despite three months of attempts to resolve the issue.

Remerciements

Je remercie Arnould Visser et Juliette Faury pour le bon déroulement de ce stage.

Table des matières

1	Introduction	1
2	Présentation du stage	3
3	Prise en Main du Lidar Velodyne VLP16	4
3.1	Configuration Initiale et Connexion	5
3.2	Acquisition des Données et Visualisation	6
3.3	Développement d'un Node ROS pour l'Exploitation des Données	7
3.4	Problèmes Techniques et Résolutions	8
3.5	Transition vers les Problèmes Logiciels	8
4	Installation de ROS Hydro : Analyse des Solutions et Approches Techniques	9
4.1	Problèmes Initiaux : Dépendances et Obsolescence	9
4.2	Solution 1 : Mise à jour des dépendances sous ROS Hydro	9
4.3	Solution 2 : Migration vers une version de ROS plus récente	10
4.4	Solution 3 : Créer un environnement de test sur une machine virtuelle	11
4.5	Solution 4 : Ancienne machine avec ROS Hydro	11
4.5.1	Copier l'environnement ROS Hydro d'une ancienne machine	12
4.5.2	Création de la clé USB bootable	13
4.5.3	Installation des bibliothèques manquantes	13
4.5.4	Installation sur le robot KUKA Youbot	13
4.5.5	Problèmes et Résolutions	14
4.5.6	Installations des différents packages	14
4.5.7	Problème restant	16
5	La découverte de la sociologie au travail	16

5.1	Vivre et Travailler à l'Étranger : Adaptation et Solitude	16
5.2	Défi de la Motivation en Face des Reculs	17
5.3	Travailler avec une Collègue en Détresse	17
5.4	Concilier Objectifs Professionnels et Personnels	18
6	Conclusion	18
	Bibliographie	20

1 Introduction

Le développement de la robotique mobile a connu des avancées significatives ces dernières années, grâce à l'amélioration des systèmes de perception et des algorithmes de navigation autonome. Dans ce contexte, le Kuka Youbot, un robot mobile à plateforme omnidirectionnelle, se distingue par sa flexibilité et ses capacités d'interaction avec son environnement. Ce stage, réalisé dans le cadre de ma deuxième année d'école d'ingénieur en robotique à l'ENSTA Bretagne, s'est déroulé à l'université Van Amsterdam, où j'ai eu l'opportunité de travailler sur ce robot.



FIGURE 1 – Laboratoire de robotique

L'objectif principal de ce stage était de rendre le Kuka Youbot autonome en déplacement, en utilisant un Lidar Velodyne VLP16 associé à une caméra et à d'autres capteurs si nécessaire. Notre projet visait à appliquer des techniques de SLAM (Simultaneous Localization and Mapping) en intégrant le Lidar pour améliorer la précision de la navigation. Ce travail s'appuie sur des recherches antérieures menées sur le même robot, qui avaient démontré la faisabilité du SLAM à l'aide d'une caméra unique [1]. Nous avons ainsi pu tirer parti des travaux d'un ancien doctorant, qui avait précédemment exploré l'utilisation d'une caméra pour réaliser du SLAM.

Au cours de mon stage, j'ai collaboré avec Juliette Faury, avec qui nous avons partagé les

responsabilités pour maximiser notre efficacité. Mon rôle principal consistait à calibrer le Lidar et à développer un node ROS pour l'acquisition et le traitement des données, tandis que Juliette s'occupait de l'intégration de la caméra et de la configuration du robot. Cependant, nous avons rencontré plusieurs difficultés techniques majeures. La principale erreur de notre part a été de ne pas faire de sauvegarde des logiciels du robot. Ainsi, lorsque nous avons dû effacer des bibliothèques du robot afin de le démarrer, nous n'avons pas pu les réinstaller facilement. En particulier, la réinstallation de ROS Hydro a été très longue et périlleuse en raison de son ancienneté. Des problèmes de compilation liés aux dépendances manquantes ont également entravé notre progression. De plus, l'intégration et la calibration du Lidar se sont révélées plus complexes que prévu, rendant la collecte et le traitement des données difficiles.

Malgré nos efforts pour résoudre ces problèmes, une erreur survenue après un mois de travail a empêché d'atteindre nos objectifs finaux, soulignant l'importance de la gestion du temps et des ressources dans un projet de développement technologique.

Cette introduction présente le cadre et les enjeux de notre projet de stage, ainsi que les objectifs que nous nous étions fixés. Dans les sections suivantes, nous aborderons plus en détail la problématique, la démarche scientifique adoptée, ainsi que les résultats obtenus et leur analyse.

2 Présentation du stage



FIGURE 2 – KUKA Youbot

Le KUKA youBot est une plateforme robotique mobile développée comme une solution open source destinée à la recherche scientifique et à l'enseignement en robotique. Il est composé d'un châssis à déplacement omnidirectionnel, pouvant accueillir un ou deux bras robotisés, ainsi qu'un PC industriel et une batterie. La communication en temps réel via EtherCAT permet un temps de réponse de 1 ms avec ses neuf entraînements, assurant une régulation précise de l'intensité, de la vitesse et de la position.

Le châssis du youBot, de dimensions compactes (53 cm de long, 36 cm de large, 11 cm de haut), est équipé de quatre roues Mecanum, permettant des déplacements omnidirectionnels, y compris latéraux et diagonaux. Un bras robotisé de 66 cm, doté d'un préhenseur à deux doigts, peut manipuler des objets de 70 mm de diamètre et jusqu'à 500 g. Les mouvements du bras sont assurés par des systèmes d'entraînement maxon, intégrant des moteurs plats et des réducteurs planétaires pour maximiser la liberté de mouvement.

Le youBot représente une avancée majeure dans le domaine de la robotique, facilitant la recherche et le développement grâce à son architecture open source basée sur Linux. Ce concept

offre des possibilités illimitées pour les chercheurs et les étudiants, rendant le youBot une plateforme de référence dans le domaine de la manipulation mobile et de l'automatisation.

3 Prise en Main du Lidar Velodyne VLP16



FIGURE 3 – Lidar utilisé

L'utilisation d'un Lidar dans les systèmes de robotique mobile, tel que le Kuka Youbot, permet d'améliorer les capacités de perception en fournissant des informations tridimensionnelles sur l'environnement. Le Velodyne VLP16, grâce à son balayage à 360 degrés, offre une résolution angulaire et une portée suffisante pour des applications de SLAM (Simultaneous Localization and

Mapping). La calibration précise de ce capteur ainsi que son intégration dans un environnement ROS (Robot Operating System) constituent des étapes essentielles pour l'exploitation des données fournies.

La prise en main du Lidar Velodyne VLP16 a nécessité plusieurs jours d'efforts, allant de la configuration initiale [2] à l'acquisition des premières données exploitables. Cette section détaille les principales étapes techniques suivies, ainsi que les défis rencontrés.

3.1 Configuration Initiale et Connexion

La première étape de la prise en main du Lidar Velodyne VLP16 a consisté à établir une connexion réseau entre le capteur et l'ordinateur. En suivant la documentation ROS dédiée au Velodyne, j'ai configuré l'adresse IP du Lidar et vérifié sa connectivité via la commande ping 192.168.1.201, qui permettait de s'assurer que le Lidar était bien détecté sur le réseau.

Suite à cela je me suis servi de l'application Veloview afin d'afficher le nuage de point que me transmettait le lidar avant de passer à Rviz.

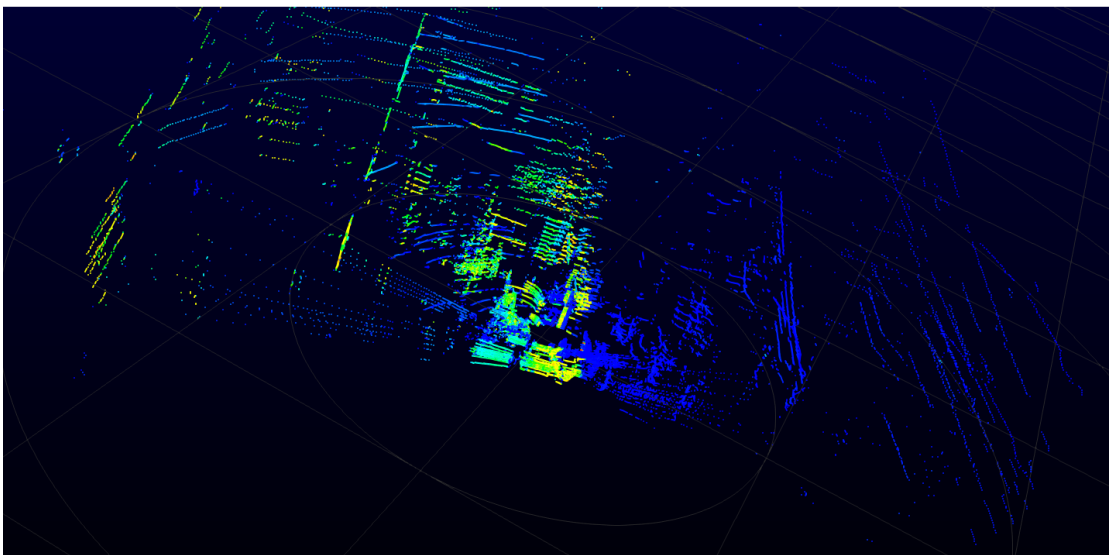


FIGURE 4 – Rendu par Veloview

Cette phase, bien que relativement simple, est cruciale pour garantir la communication entre le capteur et le système ROS. Une mauvaise configuration réseau peut entraîner des dysfonctionnements ou une absence de données. Ce premier succès a permis de passer à l'étape suivante : l'acquisition des nuages de points.

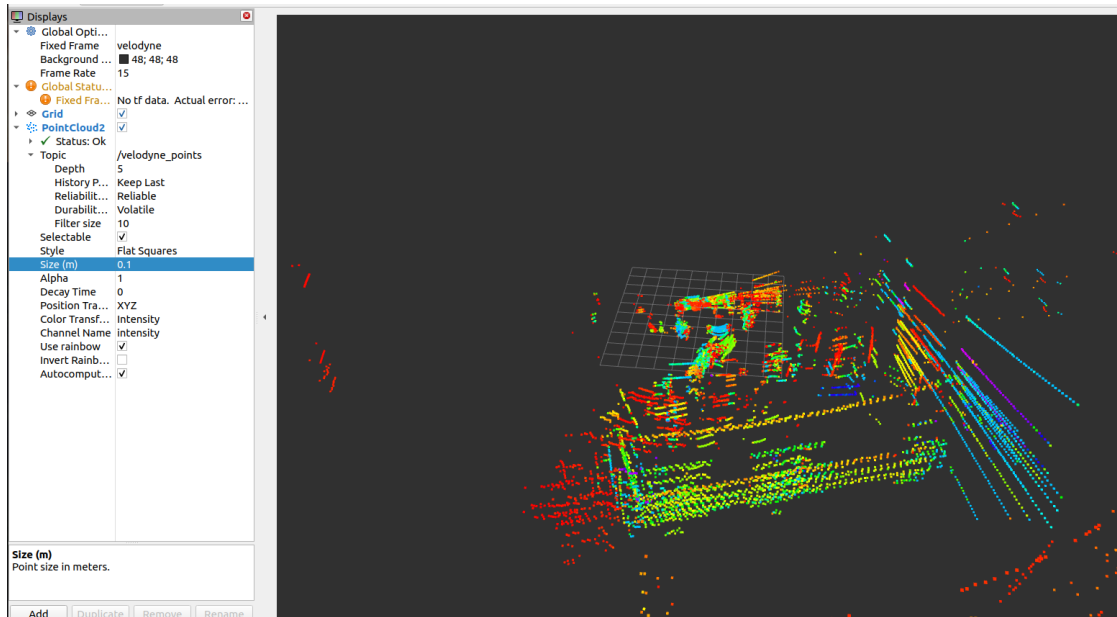


FIGURE 5 – Rendu par Rviz2

3.2 Acquisition des Données et Visualisation

La deuxième étape visait à acquérir les premières données du Lidar et à les visualiser sous forme de nuage de points. Pour cela, j'ai utilisé ROS Humble et cloné la branche `humble-devel` du dépôt GitHub du pilote Velodyne. Après avoir installé les dépendances et compilé le projet avec `colcon build`, j'ai lancé les nœuds ROS associés au Lidar avec la commande suivante :

```
ros2 launch velodyne velodyne-all-nodes-VLP16-launch.py
```

Cette commande a permis de lancer le processus de captation des données, et malgré quelques avertissements mineurs (comme l'absence de cache d'azimut ou des timeouts dans le pilote), le Lidar a fonctionné correctement. La visualisation des données s'est faite à l'aide de l'outil Rviz, où j'ai pu observer les premiers nuages de points en temps réel, ce qui a confirmé la bonne configuration du Lidar.

Cependant, plusieurs ajustements ont été nécessaires pour améliorer la visualisation. Notamment, le message d'erreur `Message Filter dropping message` dans Rviz indiquait un problème de configuration du cadre de référence (frame). Après avoir changé le cadre de référence de `map` à `velodyne` dans Rviz, le flux de données s'est stabilisé, et l'on peut alors observer le nuage de point suivant :

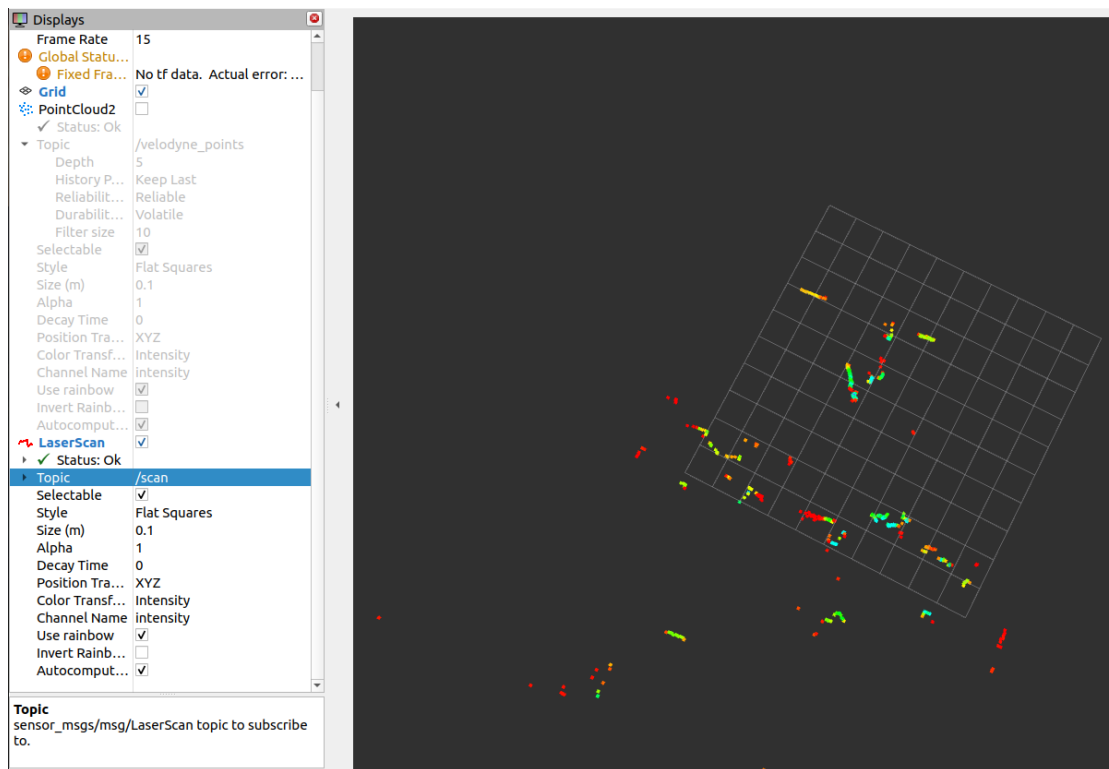


FIGURE 6 – Observation du nuage de point du node Laserscan

3.3 Développement d'un Node ROS pour l'Exploitation des Données

Afin d'exploiter les données brutes fournies par le Lidar, j'ai entrepris la création d'un node ROS dédié. Cette étape s'est concentrée sur la transformation des données du Lidar en un format plus exploitable pour notre application de SLAM. J'ai créé un package ROS en utilisant la commande suivante :

```
ros2 pkg create --build-type ament_cmake traitement_lidar --
dependencies velodyne rclpy sensor_msgs rclcpp
```

Ce package permettait de récupérer les messages publiés par le Lidar et de les convertir en données utilisables pour les algorithmes de SLAM. La principale difficulté résidait dans la gestion des nombreux messages ROS générés par le Lidar, nécessitant une optimisation pour éviter les saturations de la file d'attente de messages. C'est pour cela que j'ai choisi de prendre seulement le nuage de point que renvoyait le node `laserscan`. Dans ce node le nuage de point qui est donné est en 2D donc ce sera plus simple dans un premier temps pour l'exploitation.

3.4 Problèmes Techniques et Résolutions

Comme tout projet technique, cette phase d'intégration n'a pas été exempte de difficultés. L'un des problèmes majeurs rencontrés concernait la non-détection du Lidar par le système lors de son intégration au robot Kuka. Malgré plusieurs tentatives pour configurer le Lidar en tant que source Ethernet via `ifconfig` et des commandes telles que `dmesg | grep -i usb`, le Lidar n'apparaissait pas dans les interfaces réseau du robot.

Après avoir branché le Lidar directement sur le port Ethernet et installé les dépendances nécessaires, telles que `libpcap-dev`, j'ai finalement réussi à établir une connexion stable. Des erreurs liées aux fichiers de configuration comme `FindEigen3.cmake` ont également surgi lors de la compilation, nécessitant des ajustements dans les fichiers `CMakeLists.txt`.

3.5 Transition vers les Problèmes Logiciels

De plus, lors de l'intégration technique du Velodyne VLP16 avec le Kuka Youbot j'ai eu des difficultés liées à la gestion des dépendances logicielles. L'un des problèmes majeurs rencontrés est l'obsolescence du système ROS Hydro, utilisé par le robot, et le manque de sauvegardes ou de backups des logiciels du Youbot. Cette situation a compliqué l'installation des bibliothèques et des drivers nécessaires, notamment ceux du Lidar Velodyne. Les versions de bibliothèques, comme `libboost-all-dev`, n'étaient plus compatibles avec les versions obsolètes présentes sur le robot.

Pour résoudre ce problème, j'ai tenté de mettre à jour les bibliothèques en exécutant la commande :

```
sudo apt-get install libboost-all-dev
```

Cependant, cette mise à jour était impossible en raison de la configuration actuelle du système. En essayant de désinstaller cette bibliothèque pour installer une version plus récente, une cascade d'effets indésirables s'est produite. La commande de désinstallation suivante a également supprimé des dépendances critiques de ROS Hydro :

```
sudo apt-get remove --purge libboost-all-dev
```

À partir de ce moment, une grande partie des packages essentiels ne fonctionnaient plus correctement. Cela a déclenché une série de problèmes pour réinstaller les dépendances et remettre le système ROS en état de marche. Ces tentatives, qui se sont étendues sur plusieurs mois, se sont révélées infructueuses et ont considérablement impacté l'avancée du projet. La prochaine section détaillera les étapes que nous avons suivies pour tenter de réinstaller ROS Hydro et rétablir le fonctionnement du robot.

4 Installation de ROS Hydro : Analyse des Solutions et Approches Techniques

Suite à l'intégration initiale du Lidar Velodyne VLP16 et aux défis rencontrés avec ROS Humble, il est devenu évident que l'utilisation de ROS Hydro sur le KUKA Youbot présentait des obstacles majeurs. L'obsolescence de cette version et l'incompatibilité de nombreuses bibliothèques ont nécessité la mise en place de stratégies rigoureuses pour rétablir un environnement fonctionnel. Dans cette section, nous présentons plusieurs approches, chacune accompagnée d'une réflexion sur les résultats obtenus, en tenant compte de la démarche scientifique.

4.1 Problèmes Initiaux : Dépendances et Obsolescence

Dès le début, l'installation de ROS Hydro a été entravée par des problèmes de dépendances logicielles. Les bibliothèques critiques telles que `libpcap-dev` et `libboost-all-dev` étaient obsolètes ou difficiles à installer sur Ubuntu 12.04, le système d'exploitation requis pour ROS Hydro. Ce problème a également affecté l'intégration du Lidar Velodyne VLP16, indispensable au projet.

Les erreurs récurrentes liées à l'absence de fichiers de bibliothèques ont poussé à rechercher des solutions plus robustes. Une première analyse des erreurs a montré des incompatibilités comme celle-ci :

```
fatal error: pcap.h: No such file or directory
```

4.2 Solution 1 : Mise à jour des dépendances sous ROS Hydro

La première étape logique était de tenter de résoudre les erreurs en mettant à jour les dépendances de ROS Hydro sans changer de version. Cette approche était la plus simple dans la mesure où elle permettait de rester sur l'environnement ROS Hydro déjà en place.

Voici un exemple de commande utilisée pour réinstaller certaines bibliothèques critiques :

```
sudo apt-get update  
sudo apt-get install libpcap-dev libboost-all-dev
```

Cependant, cette solution s'est avérée insatisfaisante car la plupart des paquets requis n'étaient plus disponibles pour la distribution Ubuntu 12.04 utilisée par ROS Hydro. De plus, les versions modernes de `libboost` nécessitaient des bibliothèques obsolètes ou introuvables, entraînant une série d'erreurs de dépendances non résolues, comme en témoigne le message suivant lors de la compilation :

```
fatal error: boost/thread/lock_guard.hpp: No such file or
directory
```

Cette solution a permis de diagnostiquer l'étendue des incompatibilités mais s'est avérée insuffisante à long terme. La mise à jour manuelle des dépendances aurait été laborieuse et sujette à d'autres erreurs imprévues. La décision a donc été prise d'explorer des alternatives plus efficaces, bien que cette étape ait permis de mieux comprendre les limitations de ROS Hydro.

4.3 Solution 2 : Migration vers une version de ROS plus récente

Donc suite à ce problème d'obsolescence nous avons essayé de migrer vers une version plus récente de ROS, comme ROS Indigo ou ROS Kinetic, tout en mettant à jour le système d'exploitation vers une version plus moderne d'Ubuntu. Cette option visait à bénéficier des bibliothèques et des outils plus récents.

L'installation de ROS Indigo sur Ubuntu 14.04 a commencé par la création d'une clé USB bootable. Voici un exemple des commandes utilisées :

```
dd if=/path/to/ubuntu.iso of=/dev/sdX bs=4M
```

Une fois le système installé, ROS Indigo a été mis en place avec la commande suivante :

```
sudo apt-get install ros-indigo-desktop-full
```

Mais avant de passer à la prochaine version sur le robot, l'un des défis principaux de cette approche résidait dans la compatibilité des drivers du KUKA Youbot avec ROS Indigo. Bien que plusieurs paquets aient pu être installés sans problème, certains drivers critiques tels que `youbot_driver` n'étaient pas disponibles pour cette version de ROS. Nous avons testé cela sur machine mise à notre disposition par le maître de stage avec la configuration Ubuntu 14.04 et ROS Indigo. Pour tester cette compatibilité, nous avons utilisé la commande suivante :

```
git clone -b indigo-devel https://github.com/youbot/youbot_driver
```

Cependant, des erreurs comme celle-ci sont apparues :

```
CMake Error at youbot_driver/CMakeLists.txt:175 (add_library):
  No SOURCES given to target: YouBotDriverShared
```

Et suite à des recherches sur différents forums, une discussion avec notre maître de stage. Mais surtout avec des personnes que j'ai rencontrés lors de la robotcup24 se déroulant à Eindhoven qui avait travaillé sur le Kuka Youbot, nous avons abandonné cette piste. En effet KUKA ne pouvait

plus nous fournir les drivers compatibles avec Ros Indigo, et plusieurs des problèmes rencontrés était trop dur à résoudre. Car cela nécessiterait une réécriture complète des drivers. Les résultats étaient mitigés : bien que certains aspects fonctionnaient mieux sous ROS Indigo, les limitations des drivers du robot KUKA Youbot ont rendu cette solution impraticable à court terme.

4.4 Solution 3 : Créer un environnement de test sur une machine virtuelle

Une autre alternative efficace a été d'utiliser une machine virtuelle (VM) pour simuler l'environnement de ROS Hydro. L'objectif était de reproduire le plus fidèlement possible l'environnement ROS utilisé sur le robot, mais sur une machine plus moderne.

L'installation de ROS Hydro dans une VM a été réalisée avec la commande suivante :

```
sudo apt-get install virtualbox
vboxmanage createvm --name "Ubuntu12.04" --ostype Ubuntu_64 --
    register
vboxmanage modifyvm "Ubuntu12.04" --memory 4096 --vram 128 --cpus
    4
```

L'avantage de cette solution était qu'elle permettait de contourner les limitations matérielles du robot et de résoudre les problèmes logiciels dans un environnement contrôlé. Toutefois, des tests en conditions réelles étaient encore nécessaires pour valider les performances sur le robot lui-même.

4.5 Solution 4 : Ancienne machine avec ROS Hydro

Un espoir a surgi grâce à un ancien chercheur ayant travaillé sur le robot KUKA Youbot. J'ai rencontré ce dernier lors de mon invitation à la Robocup Eindhoven 2024.



FIGURE 7 – Robotcup Eindhoven 2024

Il a suggéré la création d'un environnement portable ROS Hydro sur une clé USB bootable, utilisant une ancienne machine fonctionnant sous Ubuntu 12.04 avec ROS Hydro, ce qui permettrait de tester les configurations et de contourner les obstacles liés à l'installation des bibliothèques manquantes sur le robot. C'est sur cette solution que j'ai donc passé le plus de temps.

4.5.1 Copier l'environnement ROS Hydro d'une ancienne machine

Une première approche pour résoudre les problèmes liés à l'installation de ROS Hydro a consisté à copier l'environnement ROS Hydro configuré sur une ancienne machine de l'université. Cette machine utilisait Ubuntu 12.04 avec toutes les bibliothèques et configurations nécessaires pour ROS Hydro. L'idée était de transférer cet environnement sur une clé USB bootable, puis de l'installer sur le KUKA Youbot pour tester et déployer les configurations sans perturber directement le robot.

Les étapes initiales consistaient à copier les fichiers ROS et les configurations critiques. L'environnement ROS Hydro était sauvegardé dans un dossier spécifique et copié sur la clé USB :

```
sudo cp -r /opt/ros/hydro /media/usb_key/
```

Cela incluait les bibliothèques critiques telles que `libboost-all-dev` et `libpcap-dev`, qui étaient

autrement difficiles à installer sur le robot directement à cause de leur obsolescence. Il était aussi essentiel de transférer le fichier `.bashrc` qui contenait les chemins des variables d'environnement pour ROS :

```
source /opt/ros/hydro/setup.bash
```

4.5.2 Création de la clé USB bootable

Une fois les fichiers copiés, il fallait rendre la clé USB bootable avec un système Ubuntu 12.04. Cette méthode a permis de tester l'environnement ROS Hydro indépendamment du robot. La création de la clé bootable a été réalisée via la commande suivante :

```
sudo dd if=ubuntu-12.04.iso of=/dev/sdX bs=4M  
sync
```

Cette approche a permis de reproduire fidèlement l'environnement ROS Hydro sur le robot tout en gardant la possibilité de réinitialiser facilement le système en cas de dysfonctionnement.

4.5.3 Installation des bibliothèques manquantes

Une fois la clé USB prête, il restait à installer les bibliothèques manquantes nécessaires au bon fonctionnement de ROS Hydro sur le robot. Ces bibliothèques incluaient `libpcap-dev` et `libboost-all-dev`, qui étaient absentes de l'environnement ROS du KUKA Youbot en raison de la difficulté d'accéder aux versions compatibles. Un exemple de commande utilisé pour installer ces bibliothèques depuis les anciens dépôts Ubuntu est le suivant :

```
sudo apt-get update  
sudo apt-get install libpcap-dev libboost-all-dev
```

Dans certains cas, il était nécessaire de forcer l'installation à partir des anciens dépôts :

```
sudo dpkg -i /path/to/libpcap-dev_2.22-1ubuntu3_i386.deb
```

Cela a permis de surmonter l'erreur suivante lors de la compilation des packages ROS :

```
fatal error: pcap.h: No such file or directory
```

4.5.4 Installation sur le robot KUKA Youbot

Une fois la clé USB prête, elle a été utilisée pour booter le KUKA Youbot. Il a fallu configurer les paramètres du BIOS pour démarrer depuis la clé USB : - Démarrer le robot et accéder au

BIOS en appuyant sur la touche F11. - Sélectionner la clé USB comme périphérique de démarrage.
- Lancer Ubuntu 12.04 à partir de la clé et vérifier la configuration réseau. Après avoir démarré avec succès, la première étape consistait à tester les commandes de base de ROS pour vérifier que toutes les bibliothèques étaient correctement installées. Voici un exemple de test de configuration réalisé après l'installation des dépendances ROS :

```
source /opt/ros/hydro/setup.bash
roscore
```

4.5.5 Problèmes et Résolutions

Malgré la réussite du démarrage de l'environnement ROS Hydro depuis la clé USB, plusieurs problèmes subsistaient. Par exemple, certaines commandes telles que `catkin_make` rencontraient des erreurs de permission :

```
bash: /opt/ros/hydro/bin/catkin_make: Permission denied
```

Pour résoudre ce problème, il a fallu changer les permissions sur les fichiers ROS installés sur le robot :

```
sudo chmod +x /opt/ros/hydro/bin/catkin_make
```

Ensuite, d'autres erreurs liées à la compilation des packages nécessaires au driver du robot.

4.5.6 Installations des différents packages

Afin de pouvoir faire des tests sur le robot nous avons besoin des packages que les drivers utilisent. Ainsi nous devons compiler le fichier `ros hydro` de la clé afin d'ensuite pouvoir l'installer. Pour cela nous faisons la commande :

```
catkin_make
```

Quand je fais cela je me retrouve avec l'erreur suivante :

```
make : *** [ cmake_check_build_system ] Error 1
Invoking " make cmake_check_build_system " failed
```

Malheureusement je ne comprends pas trop cette erreur donc pour mieux comprendre j'essaie d'installer les packages individuellement :

```
catkin_make -- only - pkg - with - deps package
```

Et ainsi j'ai des erreurs du type :

```
roscpp_core / roscpp_traits / build / devel / share /
  roscpp_traits /
cmake / roscpp_traitsConfig . cmake :165 ( find_package )
roscpp_core / roscpp_serialization / build / devel / share /
roscpp_serialization / cmake / roscpp_serializationConfig .
cmake :165 ( find_package )
-- Configuring incomplete , errors occurred !
See also "/ home / youbot / catkin_ws / build / CMakeFiles /
CMakeOutput . log ".
See also "/ home / youbot / catkin_ws / build / CMakeFiles /
  CMakeError
. log ".
make : *** [ cmake_check_build_system ] Error 1
Invoking " make cmake_check_build_system " failed
```

Ainsi je découvre que les erreurs de dépendances viennent du fait que les Cmakelist ne trouve les autres modules avec la ligne :

```
find_package ( catkin REQUIRED COMPONENTS package )
```

Donc pour essayer de contourner le fait que les Cmakelist ne trouvent pas les packages nécessaires à sa compilation je rajoute en début de fichier :

```
set ( ENV { PKG_CONFIG_PATH } "/ opt / ros / hydro / lib /
  pkgconfig ")
find_package ( PkgConfig REQUIRED )
```

Puis ensuite je commente la ligne `find_package` et le remplace par :

```
# find_package ( catkin REQUIRED COMPONENTS sensor_msgs )
pkg_check_modules ( SENSOR_MSGS REQUIRED sensor_msgs )
include_directories ( $ { SENSOR_MSGS_INCLUDE_DIRS })
link_directories ( $ { SENSOR_MSGS_LIBRARY_DIRS })
add_definitions ( $ { SENSOR_MSGS_CFLAGS_OTHER })
message ( STATUS " sensor_msgs ok //////////////////////////////////////")
```

En remplaçant `sensor_msgs` par le package nécessaire. Je fais cela pour trentaine de packages nécessaire. Et enfin pour pouvoir les installer il suffit alors d'aller dans le fichier build et de faire :

```
sudo make install package
```

Pour la quasi totalité des packages cela fonctionne. En effet pour tester cela je fais la commande `roscore` après avoir sourcer mon espace de travail et lorsque de je fais ensuite `ros topic list`. Il apparaît un noeud du style `rosout`.

4.5.7 Problème restant

Malheureusement lors du tutoriel pour tester les drivers du Youbot on doit faire compiler le module suivant à l'aide de cette commande :

```
catkin_make --only-pkg-with-deps youbot_oodl
```

Et pour ce package l'erreur persiste à cause des packages suivants :

```
youbot_oodl.cpp  
YouBotOODLWrapper.cpp  
YouBotConfiguration.cpp
```

Et quand je fais la méthode exposé précédemment j'ai une erreur :

```
- youbot_trajectory_action_server / joint_state_observer . h : No  
such file or directory  
# include < youbot_trajectory_action_server /  
joint_state_observer .h >  
- pr2_msgs / PowerBoardState . h : No such file or directory  
# include < pr2_msgs / PowerBoardState .h >
```

Et nous n'avons pas réussi à résoudre celle-ci. Les deux fichiers n'existant pas, on a pas réussi à trouver comment les installer même en essayant les méthodes montré précédemment.

5 La découverte de la sociologie au travail

5.1 Vivre et Travailler à l'Étranger : Adaptation et Solitude

Mon stage à Amsterdam a marqué ma première expérience à l'étranger, un environnement nouveau qui a nécessité un temps d'adaptation. Bien que les Pays-Bas soient un pays proche de la France, avec une culture européenne semblable et une maîtrise générale de l'anglais, certaines différences culturelles m'ont plongé dans un sentiment de décalage. Vivre hors de France, dans un contexte où je ne comprenais pas la langue locale lorsqu'elle était parlée autour de moi, a intensifié mon sentiment d'étrangeté. Ce sentiment s'est particulièrement manifesté lorsque mes collègues discutaient en néerlandais. Le fait de ne rien comprendre des conversations a créé une sorte d'isolement, même dans des moments informels comme les pauses café.

En conséquence, une solitude s'est peu à peu installée. Ce n'était pas tant le fait d'être seul physiquement, mais plutôt une sensation de déconnexion avec mon environnement immédiat. Cette situation est d'autant plus marquée par le fait que je ne pouvais pas facilement participer à des discussions légères ou impromptues avec mes collègues, qui faisaient partie de la dynamique du bureau. Des études montrent que ce sentiment d'isolement linguistique est courant chez ceux qui travaillent dans un pays où la langue locale leur est étrangère, créant une barrière supplémentaire à l'intégration sociale. Il m'a fallu du temps pour accepter cette réalité et pour trouver des moyens de la surmonter en participant plus activement aux conversations en anglais et en établissant des liens avec d'autres expatriés.

5.2 Défi de la Motivation en Face des Reculs

Une autre dimension sociologique de mon expérience a été la démotivation croissante face aux défis techniques du projet. Notre équipe travaillait sur un robot qui fonctionnait presque correctement. Cependant, une erreur dans la gestion des fichiers du système a provoqué des problèmes majeurs qui ont considérablement freiné notre progression. Ce recul a été difficile à vivre, car il a effacé des semaines de travail acharné et a créé un sentiment d'échec collectif.

L'impact psychologique de voir un projet stagner après des efforts soutenus peut être démoralisant. À mesure que les semaines avançaient sans solution immédiate, il est devenu difficile de maintenir la motivation. Notre maître de stage a reconnu cette situation et nous a conseillé de diviser nos objectifs globaux en petites missions réalisables. Selon lui, se concentrer sur des tâches simples permettait de retrouver un certain sentiment d'accomplissement. Cela a été confirmé par des théories de motivation au travail, comme celle de Locke et Latham [3], qui suggèrent que des objectifs spécifiques et atteignables permettent de maintenir l'engagement en période de stress.

En appliquant cette stratégie, nous avons commencé à retrouver de la satisfaction dans des petites victoires quotidiennes, ce qui a renforcé notre résilience collective. Cela m'a aidé à changer ma perspective sur les échecs temporaires, les voyant plutôt comme des étapes dans un processus d'apprentissage.

5.3 Travailler avec une Collègue en Détresse

Une autre dimension sociologique importante de mon stage a été la gestion des relations interpersonnelles, notamment avec une collègue sujette à des crises d'angoisse fréquentes. Ces crises ont parfois rendu le climat de travail plus lourd, affectant la dynamique de notre petite équipe. Travailler dans ces conditions nécessite non seulement de gérer son propre stress, mais aussi de se montrer empathique et de soutenir son collègue.

Il est connu que les environnements de travail où des membres d'une équipe rencontrent des difficultés psychologiques peuvent entraîner un changement dans l'atmosphère générale, parfois

provoquant un ralentissement du travail. Dans ce cas, l'équipe se retrouvait parfois paralysée pendant les périodes de crise. Cela a également modifié ma façon d'aborder les tâches quotidiennes, car il fallait régulièrement ajuster le rythme de travail pour permettre à ma collègue de se rétablir.

Bien que ces situations aient été difficiles à gérer, cela m'a appris à développer de la patience et à être plus attentif aux besoins des autres. Cela a également renforcé mon sens de la solidarité, car il était nécessaire de collaborer davantage pour avancer malgré ces obstacles.

5.4 Concilier Objectifs Professionnels et Personnels

Enfin, l'un des plus grands défis sociologiques de ce stage a été de concilier mes objectifs professionnels avec mes objectifs personnels. Contrairement à l'université, où nous avons un emploi du temps flexible, le stage exigeait un engagement complet de ma part, me laissant peu de temps pour m'investir dans mes activités personnelles. J'avais en tête un projet personnel important : me préparer pour courir un marathon, ce qui nécessitait un entraînement régulier, souvent quotidien.

Le principal défi a été de trouver un équilibre entre la fatigue physique liée à l'entraînement et les exigences mentales de mon travail. Des études montrent que jongler avec des objectifs personnels et professionnels peut parfois générer du stress, surtout lorsque les deux nécessitent une attention constante. Heureusement, mon maître de stage et ma collègue ont été compréhensifs et nous avons pu discuter ouvertement des ajustements nécessaires. Par exemple, nous avons planifié certaines périodes du stage pour être plus proactifs dans le travail, notamment durant les deux premiers mois où mon entraînement n'était pas encore intense. Puis, lorsque le marathon approchait en juillet, j'ai pu réduire légèrement ma charge de travail pour prioriser ma préparation physique.

Cette flexibilité et cette communication ouverte m'ont permis de réussir à la fois sur le plan personnel et professionnel. J'ai ainsi appris l'importance de prioriser mes objectifs et de répartir mon énergie de manière intelligente, afin d'éviter de me surcharger.

6 Conclusion

Ce stage de trois mois à l'Université d'Amsterdam, consacré au développement de l'autonomie du robot KUKA Youbot, a constitué une expérience enrichissante à la fois sur le plan technique et personnel. Bien que nous n'ayons pas atteint tous les objectifs initiaux à cause des défis rencontrés, notamment liés à l'obsolescence de ROS Hydro et à des problèmes techniques majeurs, ce projet m'a permis de développer une expertise dans l'intégration de capteurs comme le Lidar Velodyne et l'utilisation de ROS.

La dimension technique de ce stage a été marquée par la nécessité d'explorer des solutions

innovantes, comme la création d'un environnement portable avec ROS Hydro ou la migration vers des versions plus récentes de ROS. Chacune de ces approches a offert des perspectives nouvelles sur la gestion des systèmes robotiques obsolètes et sur les techniques permettant de maintenir des projets dans un cadre technologiquement limité. Bien que certaines solutions, comme la mise à jour vers ROS Indigo, aient échoué à cause des incompatibilités matérielles, elles ont fourni des leçons précieuses sur l'importance de la gestion proactive des ressources techniques.

Par ailleurs, les aspects sociologiques de ce stage m'ont confronté à des situations nouvelles et complexes. Le fait de travailler à l'étranger pour la première fois, loin de ma zone de confort, a été un défi d'adaptation. Le sentiment d'isolement causé par la barrière linguistique et culturelle a nécessité un temps d'ajustement, renforçant ma capacité à m'adapter à de nouveaux environnements. De plus, travailler sur un projet confronté à des reculs techniques a mis à l'épreuve ma motivation et mon endurance psychologique. Grâce aux conseils de mon maître de stage, j'ai appris à fragmenter mes objectifs pour retrouver de la satisfaction dans des petites avancées.

Enfin, l'équilibre entre mes objectifs professionnels et personnels, notamment avec la préparation de mon marathon, a constitué un défi supplémentaire. La gestion de cette double ambition, avec l'aide de discussions ouvertes avec mes collègues, m'a appris l'importance de la planification et de la priorisation, tant dans la vie professionnelle que personnelle.

En conclusion, ce stage m'a offert des compétences techniques essentielles dans le domaine de la robotique, mais il m'a également permis de grandir sur le plan personnel en me confrontant à des défis d'adaptation, de motivation et d'équilibre de vie. Ces apprentissages me seront précieux pour mes futurs projets en ingénierie et dans mon parcours professionnel global.

Références

- [1] Emily Mes. Visual feature detection in rtab-map with the spot robot: Applicability and performance analysis. Master's thesis, University of Amsterdam, July 2023. [1](#)
- [2] Calibration du lidar, <http://wiki.ros.org/velodyne/tutorials/getting5>
- [3] Laurent Cosnefroy. Les théories reposant sur le concept de but. Master's thesis, Dunod, 2009. [17](#)