

	 ENSTA Bretagne	ENSTA Bretagne 2, rue François Verny 29806 BREST cedex FRANCE Tel +33 (0)2 98 34 88 00 www.ensta-bretagne.fr
	Stage Assistant ROB 2025 22 octobre 2024	

Robot Explorateur



ADRIAN H. VANALLI C.

adrian.vanalli@ensta-bretagne.org

Remerciements

Je tiens à remercier Arianna Rana, qui a été quotidiennement à mes côtés pendant ces mois. Ses remarques, conseils, et son soutien inestimable et amical ont intensément contribué à mon apprentissage. Merci, Arianna, pour votre disponibilité et aide constantes.

Je tiens également à exprimer ma reconnaissance envers Antonio Petitti, qui apportait tellement d'expertise mais aussi de bonne humeur et joie même face aux difficultés pénibles. Merci, Antonio, pour votre capacité à partager vos connaissances et à me motiver à surmonter les défis rencontrés.

Mes remerciements vont également à Laura Romeo, une amie qui m'a rappelé que la qualité du travail transcende les murs du laboratoire.

Enfin, je tiens à remercier Annalisa Milella, la responsable du laboratoire, pour m'avoir accepté dans le CNR et pour l'accueil lors de mon arrivée. Merci pour votre capacité de gestion et votre reconnaissance des efforts déployés sur ce stage.

Résumé

Ce rapport présente le développement et l'implémentation d'un système de suivi de vignes pour un robot explorateur autonome dans le cadre d'un projet de recherche au laboratoire de robotique mobile du CNR en Italie. L'objectif principal était de migrer le logiciel du robot vers ROS2 afin d'améliorer la performance, la modularité et l'intégration future d'algorithmes plus complexes, tels que ceux basés sur l'intelligence artificielle. Le robot, équipé de capteurs avancés tels que la caméra Intel RealSense et un GPS à haute précision, utilise une chaîne de nodes pour analyser son environnement en 3D et ajuster sa trajectoire. Le travail inclut également une collaboration avec un autre robot, le Pioneer, pour tester des fonctionnalités en mode "swarm", dans un cadre d'agriculture de précision.

Abstract

This report presents the development and implementation of a vine-following system for an autonomous exploration robot, part of a research project at the mobile robotics laboratory of the CNR in Italy. The main objective was to migrate the robot's software to ROS2 to enhance performance, modularity, and the future integration of more complex algorithms, such as AI-based ones. The robot, equipped with advanced sensors like the Intel RealSense camera and high-precision GPS, uses a chain of nodes to analyze its 3D environment and adjust its trajectory. The work also includes collaboration with another robot, the Pioneer, to test "swarm" functionalities in a precision agriculture context.

Table des matières

Remerciements	2
Résumé	3
Abstract	3
1 Introduction	5
1.1 Context	5
1.2 Objectifs	5
2 L'Explorer	6
2.1 Hardware	7
2.1.1 NUC	7
2.1.2 Moteurs	8
2.1.3 IMU	8
2.1.4 GPS	9
2.1.5 Cameras	9
2.2 Software	9
3 Robot Operating System	11
3.1 Pourquoi migrer vers ROS2 ?	11
3.2 L'Adaptation	12
4 Collaboration avec le Pioneer	14
4.1 Reconnaissance de Couleur	15
5 Résultat final	16
6 Conclusion	16
Références bibliographiques	18

1 Introduction

Depuis des nombreuses années, l'Italie occupe des postes élevés dans le ranking des plus grands exportateurs de vins au monde. Un des plus importants contributeurs à ce succès est la région des Pouilles, dont Bari est la capitale. Il est donc évident que des améliorations dans la culture des vignes représentent une grande opportunité d'application de la robotique mobile et ont un fort potentiel de contribuer aux activités économiques de cette région.

1.1 Context

Le Conseil National de Recherche (CNR) de l'Italie compte plus de 8500 collaborateurs, dont lesquels plus de 5500 sont employés dans des activités de recherche[1]. Avec ses 228 sites et plus de 100 ans d'histoire, il s'agit d'une institution diligente et diverse, d'une importance distinguée au niveau national ainsi qu'au sein de l'Union Européenne. Il est dans ce contexte que l'institute Sistemi e Tecnologia Industriali Intelligenti per il Manifatturiero Avanzato (STIIMA) développe son expertise dans des domaines comme la robotique, la mécatronique, la fabrication et tant d'autres.

Le laboratoire de robotique mobile, à son tour, intègre le STIIMA, et est actuellement plein de projets liés à l'agriculture, y compris le Robot Explorer, qui cherche à contribuer à la transition vers l'Agriculture 4.0[2]. Tel changement envisage d'apporter des nouvelles méthodologies de gestion de la culture en intégrant des nouveaux techniques dans les tissus productifs de la filière.

1.2 Objectifs

La recherche développée dans ce stage s'inspire surtout des défis de mobilité du robot portant une variété de capteurs. L'acquisition de données est fondamentale dans l'automatisation de la gestion des vignes, une fois qu'ils fournissent des informations précieuses sur la qualité des plantes et du sol. Pour analyser la culture, qui est généralement disposée en lignes droites, le robot doit autonomement suivre les vignes en maintenant distance et orientation fixes à celle-là.

Le désir des chercheurs responsables pour le laboratoire de robotique mobile duquel ce stage se pose est d'utiliser ROS2 comme middleware, selon la vision de pouvoir appliquer des algorithmes plus lourds dans le futur, comme ceux qui sont

basés sur l'IA. Pour cette fin, il fallait utiliser un système robuste de communication entre des fonctions, comme celui que ROS2 est capable de fournir. Ce stage vise donc à migrer le logiciel du robot vers cette nouvelle distribution du système opérationnel, mais aussi, d'une manière générale, à contribuer au développement du projet où que ce soit.

2 L'Explorer

Dimensions	878x694x550 mm 34.56x37.95x21.65 in.
Maximum Payload¹	20 Kg 44 lbs
All terrains Payload	10 Kg 22 lbs
Speed	0 - 4+m/s 0 - 2,23+ mph lbs
Ground clearance	29 cm 11.41 in
Climb grade²	40°
Traversal grade²	25°
Operating time	4 Hours typical 48 hours stand-by
Battery pack	24VDC@100Ah
Battery charger	Short-circuit, over-current, over-voltage
User power	+24 VDC@5A regulated output +12 VDC@5A regulated output
Communication	RS232 up to 115200 Baud, Wi-Fi, Bluetooth, LAN
Internal sensing	1024 ppr Encoders, Voltage, current and temperature sensors

FIGURE 1 – Informations générales de l'Explorer



FIGURE 2 – Moi et mon collègue d'acier.

L'Explorer s'agit d'un véhicule électrique à quatre roues, avec chaque paire de roues attachée aux deux bras oscillants capables de se balancer indépendamment autour de ses axes [3]. Chaque roue est également capable de tourner autour de son propre axe vertical, ce qui permet de guider le robot en quatre modes différents de conduction.



FIGURE 3 – Bras oscillants

2.1 Hardware

2.1.1 NUC

Le contrôleur de bord est un NUC (CPU Intel, 8gb de RAM, HDD de 128gb) sous Ubuntu 20.04. Sur ce PC, les nodes de ROS2 Foxy sont lancés depuis un autre PC connecté via SSH.



FIGURE 4 – Intel NUC

2.1.2 Moteurs

Le robot est équipé de quatre roues, chacune avec deux moteurs. Elles sont toutes capables de tourner autour de leur axe horizontal (pour faire avancer le robot) avec un moteur DC 24V, équipé d'un encodeur 1024 pulses, mais aussi autour de l'axe vertical (selon le mode de conduction) avec un servomoteur.



FIGURE 5 – Roue

2.1.3 IMU

La XSens Mti-300 a été adoptée comme centrale inertielle. Il s'agit d'un outil d'haute qualité qui rejette des vibrations et offre des mesures d'haute précision.



FIGURE 6 – xSens Mti-300

2.1.4 GPS

Pour la localisation, le robot dispose du GPS Ublox ZED-C099-F9P. Cette carte d'application vient avec la technologie RTK intégrée et fournit des coordonnées avec une précision de l'ordre de centimètres.



FIGURE 7 – Ublox ZED-C099-F9P

2.1.5 Caméras

Les caméras Intel Realsense D435 sont très versatiles et ont des paquets prêts à être utilisés sous ROS1 et ROS2, ce qui rend pratique son utilisation. Après lancement, son node publie la pointcloud qui est utile dans notre contexte pour calculer la distance et l'orientation du robot par rapport aux plantes. Elles utilisent la stéréovision active pour capturer des images 3D en temps réel, avec une résolution de 1280x720 à 90 fps et une portée de 0,2 m à 10 m. Leur large champ de vision (86° x 57°) permet de couvrir de vastes espaces.



FIGURE 8 – Realsense D435i

2.2 Software

Le logiciel du robot est conçu spécifiquement pour accomplir la tâche de suivi de la vigne, une opération clé dans le cadre de ce projet. Pour y parvenir, chaque node joue un rôle bien défini dans le traitement des données et la prise de décision. Le point de départ de cette chaîne est la capture de données en trois dimensions via

suivre les vignes et ajuster sa trajectoire en temps réel.

Après l'acquisition et filtrage du nuage de points, le nœde appelé `wall_generator` publie la distance et l'orientation du « mur » (la ligne de vignes ou un autre obstacle que le robot doit suivre) identifié dans le topic `wall_info`. Ces informations sont filtrées par un Information Filter (le nœde `info_filter`) et envoyées au nœde de contrôle (`wallfollow`) qui, à son tour, publie la commande dans le topic `/joy_rf`. Enfin, le multiplexeur permet de changer du mode automatique (suivi de mur) au mode manuel (avec un joystick).

3 Robot Operating System

Le Robot Operating System (ROS) est un middleware, à savoir un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques[4], créé par la société américaine Willow Garage en 2007. Aujourd'hui, ce logiciel est utilisé dans des robots appliqués dans de nombreuses activités différentes. Ses principes (peer to peer, basé sur des outils, multi langages, léger, gratuit et open source) ont fait de lui une excellente option pour l'Explorer.

3.1 Pourquoi migrer vers ROS2 ?

Selon Florian Cantori, ingénieur chez Awabots, « l'ambition de ROS2 est tout simplement de maintenir tout ce que ROS1 a fait de bien, en améliorant la fiabilité du système et en le préparant à une application industrielle plus large et plus robuste » [5]. À cette fin, il ne s'agissait pas de simplement mettre à jour l'ensemble de logiciels offerts par ROS mais de créer un nouveau système avec de nouveaux principes surpassant la version précédente.

L'un des changements majeurs introduits par ROS2 est l'intégration d'une architecture de communication basée sur DDS (Data Distribution Service), plus robuste et flexible que les protocoles propriétaires utilisés par ROS, comme le TCPROS, permettant la communication en temps réel. Cet améliorement rend ROS2 applicable dans les contextes industriels et médicaux, par exemple, vu qu'il améliore la sécurité, l'escalabilité et l'interopérabilité des systèmes robotiques, notamment dans des environnements distribués ou contraints en ressources. De plus, ROS2 est maintenant compatible avec plusieurs plateformes, telles que Linux, Windows, et macOS, offrant

ainsi une plus grande flexibilité aux développeurs et aux entreprises utilisant des écosystèmes variés.

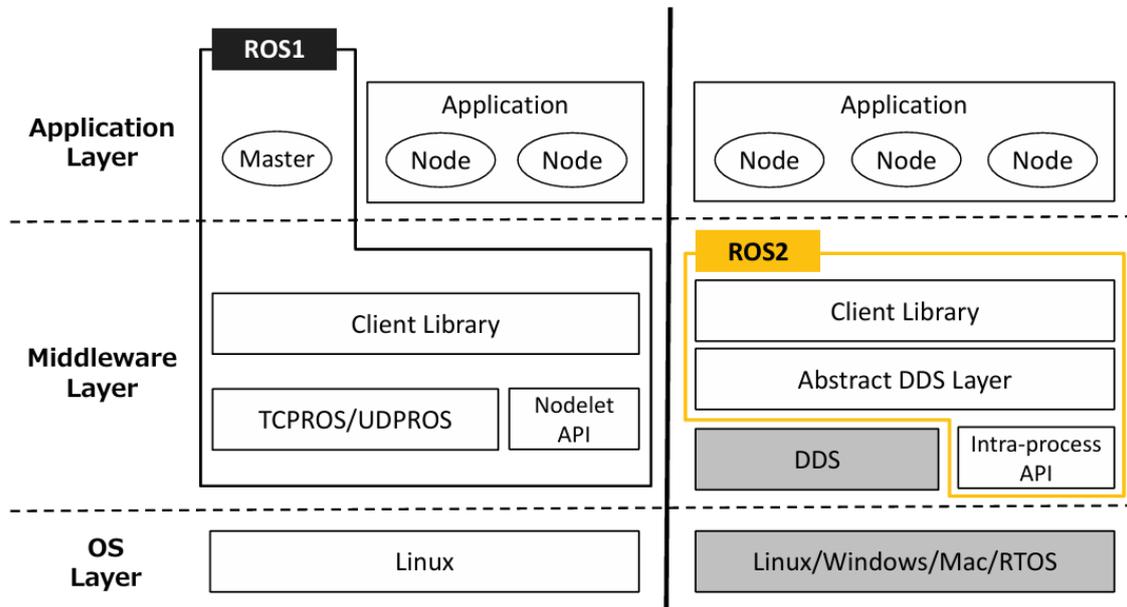


FIGURE 10 – Source : « Exploring the performance of ROS2 »[6]

La modularité est aussi un grand avantage pour ceux qui migrent leur système vers ROS2. Sans le concept de roscore, le réseau de nœuds est devenu beaucoup plus dynamique et sûr. La décentralisation, dans un premier moment, peut ne pas sembler une grande évolution forcément, mais elle facilite la maintenance, les mises à jour et le déploiement lorsque notre système robotique devient complexe[7].

Finalement, sachant que la fin de vie et de support de la version ROS1 Noetic a été annoncée pour mai 2025, le travail mené au sein de ce stage est incontournable. Pourtant, ça assure également encore quelques années pour ROS, dans lesquelles il sera souvent le cas de travailler parallèlement avec ces deux versions. Afin de faciliter cette transition, une cohabitation est possible par l'interconnexion des outils ROS et ROS2 via des 'ponts' (bridges), qui ont été largement utilisés dans ce stage.

3.2 L'Adaptation

Une fois que le fonctionnement du système existant a été compris, une stratégie de migration a été progressivement formulée. D'abord, il fallait mettre en place le lancement de la caméra Intel RealSense, ce qui a été facilité par l'existence

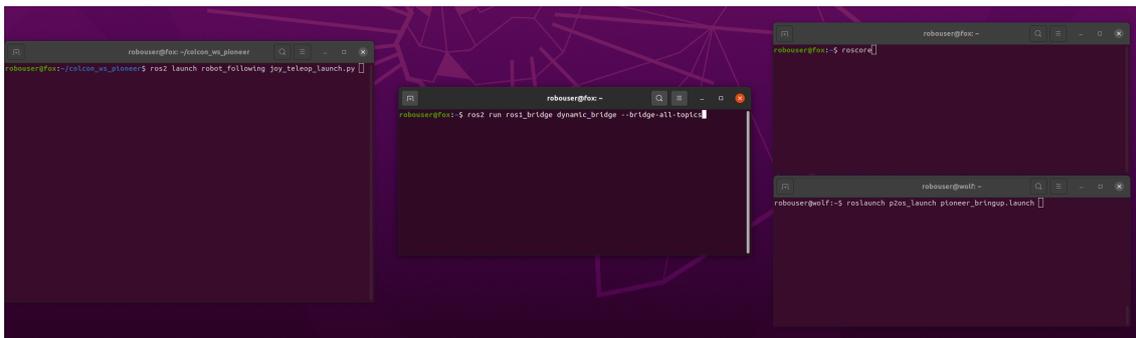


FIGURE 11 – Utilisation de bridge avec le Pioneer

d'un package compatible, quasiment plug-and-play, pour ROS2. Ce package, déjà largement utilisé dans ROS1, a permis d'intégrer la capture de données 3D sans grande difficulté, vu qu'il disposait de plusieurs nodes pour travail avec cet instrument, notamment pour le traitement du pointcloud.

Cependant, un défi majeur est apparu lorsqu'on a noté que le traitement des nuages de points était géré via un nodelet, une technologie permettant de lancer plusieurs nodes dans le même processus afin de réduire les surcoûts liés à la communication inter-processus. Dans ROS2, on n'utilise plus ce concept de nodelet; il a été remplacé par les Composable Nodes. Ils sont plus légers et plus flexibles que les nodelets, bien que leur implémentation nécessite une adaptation des pratiques de développement, car ils introduisent une architecture plus "primitive" que nous n'avions pas abordée en cours.

Pour la plupart des nodes, des modifications plus générales ont dû être apportées, comme pour les bibliothèques et structures de base :

- L'importation de `rclpy` en lieu et place de `rospy`, car ROS2 utilise une nouvelle API Python plus performante et mieux structurée;
- Sous ROS2, la classe doit hériter de `rclpy.node.Node`, et la gestion des publications et souscriptions se fait différemment, en utilisant des objets `Publisher` et `Subscriber` adaptés à ROS2;
- la fonction `main` a subi des changements pour ROS2, où l'initialisation et l'exécution du node passent désormais par `rclpy.init()` et `rclpy.spin()`, une différence notable par rapport à la structure de ROS1.

Également, tous les launch files précédents, écrits en XML, ont été remplacés dans ROS2 par des scripts python. J'ai profité de cette transition pour introduire davantage

de modularité dans les fichiers de lancement, en incluant des paramètres configurables et des structures conditionnelles pour améliorer l'efficacité et l'adaptabilité du système. Par exemple, une option pour lancer une ou deux caméras, ainsi que l'option de changer dynamiquement les types de filtrage.

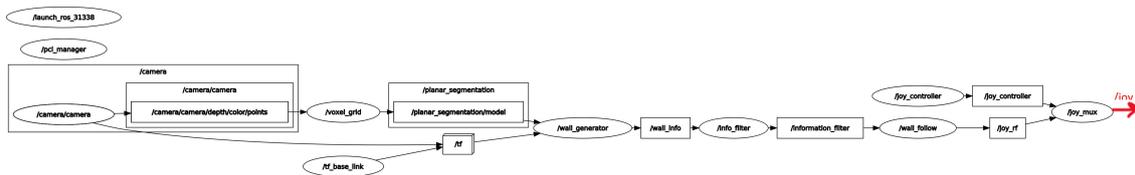


FIGURE 12 – Une des propositions du système en ROS2

4 Collaboration avec le Pioneer

Le Pioneer est un robot tout-terrain 4x4 équipé d'un système de direction de type skid-steering, ce qui signifie qu'il peut tourner en modifiant la vitesse des roues de chaque côté, sans nécessiter de direction active. Il a été introduit pour la première fois dans le laboratoire de robotique mobile il y a environ dix ans, où il a servi à divers projets de recherche et de développement. Malgré ses capacités robustes, le Pioneer était inactif depuis plusieurs années avant ce stage. Il est équipé d'un PC embarqué fonctionnant sous Linux 12, avec ROS Fuerte comme middleware, une version de ROS qui était couramment utilisée à l'époque de son lancement. La communication avec le robot se fait via SSH, en utilisant un réseau Ethernet, permettant de contrôler et de surveiller ses opérations à distance.

Au dernier mois de stage, Antonio, l'un des chercheurs responsables du laboratoire, a proposé de réactiver ce robot afin de l'intégrer dans les travaux actuels du laboratoire. Son idée était d'exploiter les capacités du Pioneer en le faisant collaborer avec le robot Explorer sous une logique de « swarm ». Cette approche permet à plusieurs robots de travailler ensemble de manière coordonnée, partageant des informations et des tâches pour accomplir des missions plus complexes. Afin de rendre cela possible, une interconnexion entre les deux systèmes a été réalisée en utilisant ROS2 pour l'Explorer et un système de bridge entre ROS Fuerte et ROS2 Foxy. Ce projet de collaboration a permis non seulement de redonner vie au Pioneer, mais aussi de tester et valider des concepts innovants de robotique en swarm, ouvrant la voie à des

applications dans des domaines variés dans le futur de ce laboratoire, notamment l'agriculture de précision et la gestion collaborative de flottes de robots.



FIGURE 13 – Pioneer

4.1 Reconnaissance de Couleur

Face aux différentes options pour la co-localisation du Pioneer par rapport à l'Explorer, on a choisi d'utiliser un algorithme de reconnaissance de couleur, grâce à sa simplicité et sa fonctionnalité. Avec OpenCV, le Pioneer était censé voir la couleur rouge de l'Explorer et le suivre. À cette fin, deux scripts ont été élaborés à partir des exemples trouvés en ligne : un pour calibrer la couleur et l'autre qui donne la distance aux objets détectés.

Le script de calibration des couleurs utilise OpenCV pour trouver les meilleures valeurs de teinte (H), saturation (S) et valeur (V) pour détecter le rouge, caractéristique de l'Explorer, avec des barres de défilement interactives permettant d'ajuster ces paramètres en temps réel. Une fois les seuils définis, le script applique un filtre sur l'image vidéo, isolant les zones rouges. Ensuite, une transformation morphologique est utilisée pour nettoyer l'image (dilatation), une technique étudiée en classe, avant de détecter les contours des objets rouges présents. Les contours détectés sont ensuite utilisés pour encadrer l'Explorer dans l'image et fournir ses coordonnées, facilitant ainsi son suivi par le Pioneer.

Dans le second script, le pipeline `pyrealsense2` permet de récupérer des images de profondeur et des images couleur synchronisées. Après détection de la couleur rouge, le script calcule la distance entre le Pioneer et l'Explorer en utilisant les coordonnées du centre de l'objet détecté. La distance est estimée en mesurant la profondeur à l'aide du capteur de profondeur de la caméra. Si l'Explorer est trop éloigné (au-delà de 3,5 mètres), un message est affiché indiquant que l'objet est hors de portée.

5 Résultat final

Habitué à la méthode Agile de développement, j'avais prévu, même avant le début de la moindre modification, la réalisation de vérifications intermédiaires. À cette fin, avec le soutien des bridges, on augmenterait progressivement la proportion de ROS2/ROS utilisée par le système. Tout d'abord, la "coupe" serait effectuée juste après l'acquisition du nuage de points, donc au niveau du topic `"/camera/depth/color/points"`. Ensuite, la partie de filtrage serait également migrée vers ROS2 et la coupe se ferait au niveau du topic `"/wall_info"`, et ainsi de suite.

À la fin, l'Explorer a réussi à suivre un mur de manière totalement autonome, en opérant sous ROS2. Pourtant, il n'a malheureusement pas été possible de vérifier cette fonctionnalité dans les vignes, due à plusieurs facteurs. En raison de ses dimensions, déplacer le robot était coûteux, et les déplacements vers les fermes étaient donc peu fréquents. De plus, au cours du dernier mois de stage, une mauvaise initialisation du robot a provoqué la rupture d'un câble d'importance cruciale, ce qui a également joué un rôle important dans la décision de travailler avec le Pioneer.

6 Conclusion

Intégrer le laboratoire de robotique mobile au sein d'une institution comme le CNR a été une expérience très enrichissante à plusieurs égards. Du côté humain, le travail en équipe, l'autogestion, la vision critique, la planification en autonomie, ainsi que les échanges culturels et linguistiques ont été présents tout au long de ces mois. Ce stage m'a donné l'opportunité de renforcer ma capacité à garder l'esprit ouvert face aux différentes approches pour résoudre les enjeux techniques et logistiques que l'ingénierie nous présente. De la même façon, le travail en équipe m'a

beaucoup encouragé à bien élaborer et argumenter les solutions que j'avais formulées. L'autonomie a aussi été stimulée dans plusieurs situations, où des responsabilités m'ont été attribuées, tout en tenant compte de la gravité de chaque problème.

Pareillement, sur les plans théorique et pratique, ce stage marque très positivement ma trajectoire professionnelle. La migration du logiciel de l'Explorer vers ROS2 a permis d'apporter des améliorations significatives en matière de modularité, de performance et de préparation pour des algorithmes plus avancés, notamment ceux basés sur l'intelligence artificielle. Malgré quelques défis rencontrés, notamment lors de l'intégration des Composable Nodes et des ajustements nécessaires dans les launch files, la transition vers ROS2 est sur la bonne voie.

La collaboration avec le robot Pioneer a ouvert de nouvelles perspectives, en explorant des concepts innovants comme la robotique en swarm, qui est aussi un intérêt personnel à moi. Bien que certaines limitations matérielles aient empêché une validation complète du système dans les vignes, les résultats obtenus nous permettent d'être optimistes sur la solution ROS2.

En conclusion, participer à cette expérience en étant capable de contribuer aux avancements techniques de l'Explorer a été absolument satisfaisant et gratifiant. Ce projet est un atout important avec plein de potentiel dans la transition vers l'Agriculture 4.0.

Références bibliographiques

- [1] <https://www.cnr.it>, juin de 2024.
- [2] <https://www.stiima.cnr.it>, juin de 2024.
- [3] <https://www.robo-dyne.com/explorer-4wd-ugv-four-wheel-driving-steering-robot-vehicle/>, juillet de 2024.
- [4] <https://fr.wikipedia.org/wiki/Middleware>, septembre de 2024.
- [5] <https://awabot.com/en/ros1-ros2-differences-en/>, septembre de 2024.
- [6] Yuya Maruyama, Shinpei Kato, Takuya Azumi - « Exploring the Performance of ROS2 », 2016.
- [7] <https://design.ros2.org/articles/changes.html>, septembre de 2024.