

# Bureau d'étude véhicule autonome

- ENSEM -

L'ENSEM a récemment fait l'acquisition de plusieurs robots *agilex Limo* (représentés sur la figure 1). Malheureusement les fonctionnalités logicielles fournies avec ce robot ne lui donnent pas entière satisfaction. Nous avons donc mandaté votre bureau d'étude pour réaliser et évaluer le déplacement autonome et en toute sécurité du robot *agilex Limo*, représenté sur la figure 1.



FIGURE 1 – Robot agilex Limo

## 1 Cahier des charges

Il est demandé que les robots réalisent les deux fonctionnalités principales suivantes : (i) le déplacement autonome du robot entre deux points donnés et (ii) la supervision du robot afin d'analyser les performances du robot.

L'échéance du projet est fixée au *23 janvier 2024*. Une présentation de votre travail et de vos résultats sera réalisée ce même jour, suivie d'une démonstration pratique sur les robots. Chaque groupe présentera ses résultats à l'aide d'un powerpoint en *20 minutes*. Votre programme final devra également être livré. Nous souhaitons un développement en *Python/ROS*. Une attention particulière sera portée sur la lisibilité et la structure de votre livrable.

### 1.1 Etape 1 : algorithme de déplacement autonome

Dans le plan  $(x, y)$ , nous souhaitons maîtriser l'évolution de la *position* du robot  $(x_R, y_R)$  mais aussi son *orientation* par rapport à l'axe  $x$ , que nous notons  $\theta_R$ . A partir de n'importe

quelle position et orientation initiales du robot, notée  $(x_0, y_0, \theta_0)$ , vous devez concevoir et mettre en œuvre un algorithme de commande qui devra l’amener vers une position *et* une orientation cible, notée  $(x^*, y^*, \theta^*)$ .

Afin de nous convaincre d’utiliser votre algorithme, nous demandons :

- des garanties *théoriques* de fonctionnement ;
- des validations *en simulation* ;
- des validations *expérimentales* sur le robot.

Pour pouvoir appliquer cet algorithme et tester ces premiers type de déplacement, il faut pour cela récupérer les différentes données des capteurs pour obtenir la position de la voiture, et ensuite concevoir en conséquence la commande nécessaire pour déplacer le véhicule.

## 1.2 Etape 2 : supervision du véhicule

Le robot génère des données issues de ses capteurs et de ses programmes internes. Nous souhaitons pouvoir récupérer et afficher ces données à distance sur un poste dit de *supervision*. Un autre aspect de cette fonction concerne la commande et la planification de tâches à effectuer par le robot. Nous souhaitons également pouvoir envoyer des instructions au robot via cette fonction de supervision (e.g. démarre, stop, action 1, ...).

Ainsi, nous vous demandons de :

- Visualiser la position  $(x, y, \theta)$  du véhicule dans l’espace
- Utiliser Rviz et RQt pour visualiser les capteurs, et positionner les obstacles autour du robot
- Créer une interface de votre choix (e.g. Python) pour commander le robot selon des instructions simples avec ROS

Pour réaliser ces fonctions, vous aurez besoin de comprendre le fonctionnement des capteurs qui équipent le robot, mettre en œuvre un nœud ROS 2, programmer une interface.

## 2 Ressources

Afin de vous aider dans la réalisation de ce projet, nous avons mis à votre dispositions plusieurs ressources.

**Experts.** Tout au long du projet, vous avez à votre disposition une équipe de quatre experts suivants pour répondre à vos questions :

- *Abdelkader Lahmadi* ; maître de conférence à l’université de Lorraine ; compétences : informatique ; mail : [abdelkader.lahmadi@univ-lorraine.fr](mailto:abdelkader.lahmadi@univ-lorraine.fr) ; bureau : 203 Rouge
- *Amaury Saint-Jore* ; doctorant au LORIA ; compétences : informatique, robotique ; mail : [amaury.saint-jore@univ-lorraine.fr](mailto:amaury.saint-jore@univ-lorraine.fr) ; bureau : 202 Rouge
- *Clément Fauvel* ; ingénieur de recherche à l’université de Lorraine ; compétences : informatique, automatique ; mail : [clement.fauvel@univ-lorraine.fr](mailto:clement.fauvel@univ-lorraine.fr) ; bureau : 108 Bleu
- *Jérémy Kreiss* ; maître de conférence à l’université de Lorraine ; compétences : automatique ; mail : [jeremie.kreiss@univ-lorraine.fr](mailto:jeremie.kreiss@univ-lorraine.fr) ; bureau : 122 Rouge

*Modalité de l’aide* : Vous pouvez prendre contact avec eux par email pour fixer une rencontre.

**Formations.** Nous avons identifié deux thématiques essentielles pour réaliser ce projet : (i) la commande de véhicules autonomes et (ii) la programmation sous ROS (Robot Operating System). Afin de vous fournir les compétences de base sur ces thématiques, nous vous avons inscrit à :

- une formation de 4H le *vendredi 29/09/2023 de 8H à 12H* concernant la partie commande ;
- et une formation de 8H les *mardi 03/10/2023 et vendredi 06/10/2023 de 8H à 12H* concernant la prise en main de ROS.

Grâce à la formation sur la commande de véhicules autonomes, vous obtiendrez les compétences suivantes :

- la connaissance des principaux types de véhicules sur roues ;
- la capacité à modéliser un véhicule de type voiture ;
- la capacité à concevoir une loi de commande pour déplacer un véhicule de type voiture de manière autonome ;
- la capacité à analyser vos lois de commandes afin d’obtenir certaines garanties théoriques ;
- la capacité à simuler un véhicule de type voiture et sa commande et à interpréter les simulations.

Grâce à la formation concernant la programmation sous ROS et Python, vous obtiendrez les compétences suivantes :

- Rechercher et utiliser les documentations et les bibliothèques concernant un produit (ici notre robot LIMO) ;
- Comprendre les avantages et inconvénients d’un middleware comme ROS 1 et 2 ;
- Rappels et usages de Linux ;
- Comprendre les principes de bases de ROS et le prendre en main rapidement ;
- Afficher et manipuler les différents objets propres à ROS ;
- Programmer en Python des nodes, topics, paramètres, etc ;
- Utiliser des outils (Rviz et RQt) pour visualiser les données dans l’espace en 3D, afficher les informations du système, tracer des courbes, etc.

**Documentation.** Les différents documentations sont accessibles via les liens suivants :

- [Présentation site officiel AgileX LIMO](#)
- [GitHub documentations officielles LIMO](#)
- [Documentation sur AgileX LIMO \(fournisseur Trossen Robotics\)](#)
- [documentation ROS 1 \(Eloquent\)](#)
- [documentation ROS 2 \(Humble\)](#)
- [documentation Nav2](#)
- [documentation Rviz](#)

### 3 Organisation

Pour ce projet, vous devez vous répartir en équipes de *3 personnes*.

Il vous est demandé pour le vendredi 29/09/2023 de découper votre projet sous formes de tâches élémentaires et d’en faire une répartition entre les membres du bureau d’étude. Un planning prévisionnel est également à fournir.