# Rapport de la Phase de Simulation : Conception d'une Flotte de Fauteuils Roulants Autonomes

### Introduction

Le développement d’une flotte de fauteuils roulants autonomes représente une avancée technologique majeure pour améliorer la mobilité et l’autonomie des personnes à mobilité réduite. Afin de garantir la faisabilité et la performance d’un tel système, une phase initiale de simulation a été réalisée. Cette étape visait à valider les principes fondamentaux de suivi et de coordination entre deux robots, un robot suiveur et un robot meneur, en exploitant des technologies avancées comme le LIDAR, la vision par caméra et les codes ArUco.

La simulation a été conçue pour recréer un environnement réaliste, incluant des trajectoires complexes, des obstacles, et des scénarios dynamiques. Les outils utilisés, **CoppeliaSim** et **MATLAB**, ont permis d’intégrer et d’évaluer les performances des capteurs embarqués, ainsi que d’analyser les données en temps réel. Cette approche a posé les bases solides pour garantir que les concepts testés dans un environnement virtuel puissent être transposés efficacement au monde réel.

Les résultats de cette phase de simulation ont confirmé la capacité du robot suiveur à maintenir un suivi précis et stable du robot meneur, ouvrant ainsi la voie au prototypage physique. Cette étape clé marque une avancée importante vers la concrétisation d’un système de mobilité autonome fiable et adaptable.

### Objectifs de la Simulation

L'objectif principal de cette phase était de développer et valider un système de suivi autonome dans lequel :

**Suivi autonome précis** :

* Maintien d'une distance constante (0.5 mètre) entre le robot suiveur et le robot meneur.
* Alignement directionnel et adaptation dynamique à la trajectoire.

**Intégration des technologies de perception** :

* Utilisation du LIDAR pour mesurer les distances et détecter les obstacles.
* Exploitation de la caméra et des ArUco codes pour localiser et orienter le robot suiveur par rapport au meneur.

**Scénarios dynamiques** :

* Tests sur des trajectoires variées (courbes, zigzags, obstacles).
* Adaptation à des variations de vitesse du meneur du robot.

**Validation du contrôleur bas niveau** :

* Évaluation des performances du contrôleur pour ajuster les vitesses des roues en fonction des écarts mesurés par les capteurs et maintenir une trajectoire stable.

**Synchronisation des outils** :

* Coordination fluide entre CoppeliaSim et MATLAB pour garantir des ajustements en temps réel et analyser les performances.

### Approche Méthodologique & Outils Utilisés

Pour réaliser cette simulation, les outils suivants ont été employés :

* **CoppeliaSim** :
  + Configuration des robots (dimensions, capteurs, contrôleurs).
  + Paramétrage de l'environnement, comprenant la piste de déplacement, les obstacles, et le ArUco code placé sur le robot meneur.
  + Simulation des trajectoires de déplacement des robots dans un environnement contrôlé.
* **MATLAB** :
  + Analyse des données collectées depuis CoppeliaSim, notamment les mesures de distance et d'alignement.
  + Implémentation du contrôleur bas niveau qui ajuste les vitesses des roues en fonction des écarts détectés.
* **Technologies embarquées** :
  + **LIDAR** : Permet de mesurer la distance entre le robot suiveur et les objets environnants, incluant le robot meneur, grâce à un balayage laser. En comparant les distances enregistrées sur plusieurs angles, il détermine la position relative du robot meneur.
  + **Caméra et ArUco code** : La caméra montée sur le robot suiveur capture l'image du ArUco code placé à l'arrière du robot meneur. Cette lecture permet :
    - De calculer précisément l'écart latéral entre les robots en analysant la position du ArUco code dans le champ de vision.
    - D'obtenir l'orientation du robot suiveur par rapport au meneur en déduisant l'angle de rotation du ArUco code par rapport à l'axe de déplacement.

### Configuration de la Simulation

La simulation s'est déroulée en plusieurs étapes détaillées :

1. **Configuration initiale des robots :**
   * Le robot meneur a été équipé d'un ArUco code placé à l'arrière, servant de repère visuel pour le robot suiveur.
   * Le robot suiveur a été équipé d'un LIDAR pour scanner l'environnement et d'une caméra orientée vers le ArUco code pour effectuer les lectures nécessaires.
   * Les capteurs ont été calibrés pour assurer des mesures fiables dans un environnement de simulation. Le LIDAR a été configuré pour détecter des objets dans un rayon de 3 mètres, tandis que la caméra a été testée pour lire des ArUco codes à différentes distances et angles.
2. **Création de l'environnement simulé :**
   * Une piste virtuelle a été conçue dans CoppeliaSim pour représenter un environnement réaliste. Cette piste comportait des segments droits, des virages serrés et des obstacles placés stratégiquement.
   * Les obstacles ont été ajoutés pour tester les capacités du LIDAR à détecter les objets environnants et permettre au robot suiveur d'ajuster sa trajectoire en conséquence.
3. **Paramétrage des trajectoires :**
   * **Trajectoire du robot meneur :** Le robot meneur a été programmé pour suivre une trajectoire complexe, composée de courbes, de zigzags et de lignes droites. Ces variations permettent de tester la capacité du robot suiveur à adapter sa vitesse et sa direction en temps réel.
   * **Mécanisme de suivi du robot suiveur :**
     + Le robot suiveur utilise les données du LIDAR pour détecter la distance jusqu'au robot meneur et éviter les obstacles. Le capteur envoie des informations sur l'angle et la distance au contrôleur bas niveau.
     + La caméra du robot suiveur capture les images du ArUco code à intervalles réguliers. Les données extraites incluent la position et l'orientation relatives du ArUco code, utilisées pour corriger la trajectoire.
   * **Optimisation des mouvements :** Les vitesses linéaires et angulaires ont été ajustées pour garantir une transition fluide entre les segments droits et les courbes. Le robot suiveur a été programmé pour ne jamais dépasser une vitesse maximale définie, tout en maintenant un écart constant avec le robot meneur.
4. **Analyse et traitement des données :**
   * Les données brutes de distance et d'alignement obtenues via le LIDAR et la caméra ont été collectées en temps réel et envoyées à MATLAB.
   * **Traitement dans MATLAB :**
     + Les données du LIDAR ont été utilisées pour connaître la distance du robot situé à l’avant, permettant au robot suiveur de maintenir une distance de sécurité.
     + Les images capturées par la caméra ont été analysées pour extraire les informations du ArUco code. Ces informations incluaient l'angle d'orientation et l'écart latéral, essentiels pour ajuster la direction du robot suiveur.
   * Les résultats des ajustements ont été visualisés sous forme de graphiques montrant la distance et l'alignement entre les deux robots au cours du temps. Ces visualisations ont permis de valider la stabilité et la précision du système de suivi.

### Résultats Obtenus

Les résultats de la simulation ont permis de valider les hypothèses initiales et de démontrer les capacités du système de suivi autonome. Ils sont résumés comme suit :

**5.1. Suivi Précis et Stable**

* Le robot suiveur a maintenu un écart constant de 0.5 mètre avec le robot meneur sur l’ensemble des scénarios simulés, incluant des trajectoires complexes et des variations de vitesse.
* L’alignement directionnel a été assuré grâce à l’utilisation combinée des données de la caméra et du LIDAR, permettant des ajustements en temps réel sans oscillations significatives.

**5.2. Fiabilité des Capteurs**

* **LIDAR :**
  + Le capteur LIDAR a détecté avec précision les obstacles et les distances entre les deux robots, garantissant un suivi continu et sans collision.
  + Les variations rapides dans l’environnement n’ont pas affecté la fiabilité des mesures, grâce à une calibration initiale réussie.
* **Caméra et ArUco code :**
  + La caméra a capturé les images du ArUco code avec une précision élevée même avec des angles d’inclinaison modérés.
  + Les données extraites du ArUco code ont permis une localisation fine et une orientation précise du robot suiveur par rapport au meneur.

**5.3. Performance du Contrôleur bas niveau**

* Le contrôleur bas niveau a permis d’ajuster rapidement la trajectoire du robot suiveur en minimisant les erreurs de distance et d’alignement.
* Les paramètres optimisés du contrôleur bas niveau ont limité les oscillations et garanti une réponse fluide lors des changements de direction ou de vitesse du robot meneur.

**5.4. Visualisation des Données**

* Les données collectées en temps réel ont été analysées dans MATLAB et présentées sous forme de graphiques illustrant la vitesse linéaire des robots.
* Ces visualisations ont confirmé la robustesse et la stabilité des algorithmes de contrôle utilisés.

**5.5. Robustesse Globale du Système**

* La simulation a démontré que le système est capable de s’adapter à des scénarios dynamiques complexes, incluant des obstacles imprévus, des changements de vitesse, et des trajectoires sinueuses.
* L’intégration des technologies et des outils a permis une exécution fluide et coordonnée, confirmant la faisabilité du passage à une phase de prototypage physique.

En conclusion, les résultats obtenus valident les concepts testés dans l’environnement simulé et posent les bases pour le développement d’un prototype physique capable d’opérer dans un environnement réel.

### Difficultés Rencontrées

La réalisation de la simulation a révélé plusieurs défis techniques et opérationnels, nécessitant des ajustements spécifiques à différents niveaux :

**6.1. Calibrage des Capteurs**

* **ArUco Code et Caméra :**
  + Le seuil de détection du ArUco code a demandé plusieurs ajustements pour éviter les erreurs de lecture dues aux variations de distance ou d’angle. Les tests initiaux ont révélé des échecs fréquents de reconnaissance lorsque la caméra s’éloignait au-delà de 1,5 mètre.
  + La calibration de la caméra pour garantir des lectures fiables à toutes les distances et sous différents angles a été laborieuse.
* **LIDAR :**
  + La précision du LIDAR s’est avérée sensible aux variations rapides de l’environnement, comme des mouvements soudains ou des objets mobiles à proximité. Cela a nécessité des ajustements pour filtrer les données bruitées sans compromettre la rapidité des calculs.

**6.2. Problèmes de Traitement d’Images**

* **Temps de Traitement :**
  + L’analyse des images capturées par la caméra était initialement trop lente pour permettre des ajustements en temps réel. Les délais accumulés ont affecté la réactivité du robot suiveur face aux changements rapides de trajectoire du robot meneur.
* **Erreurs de Lecture :**
  + Les erreurs dans l’identification des ArUco codes ont été fréquentes lorsque ces derniers étaient inclinés ou partiellement masqués. Cela a conduit à des ajustements dans le positionnement du ArUco code sur le robot meneur et dans l’angle de la caméra sur le robot suiveur.

**6.3. Synchronisation entre Outils**

* **Interface CoppeliaSim-MATLAB :**
  + L’échange de données en temps réel entre CoppeliaSim et MATLAB a été difficile à mettre en place.

**6.4. Ajustements des Paramètres du Contrôleur bas niveau**

* **Instabilités Initiales :**
  + Les paramètres initiaux du contrôleur bas niveau ont conduit à des oscillations importantes dans les trajectoires du robot suiveur, notamment lors des changements rapides de direction ou de vitesse.
  + Plusieurs itérations ont été nécessaires pour équilibrer les composantes proportionnelles afin d’obtenir une réponse fluide.

En résolvant ces difficultés, la simulation a non seulement permis d’affiner les outils et les méthodes utilisées, mais également de mieux comprendre les défis qui pourraient survenir lors du prototypage physique.

### Conclusion

La phase de simulation a permis d'atteindre les objectifs fixés :

* Le robot suiveur a pu suivre avec précision le robot meneur tout en maintenant un écart constant et une bonne orientation.
* Les outils CoppeliaSim et MATLAB ont démontré leur complémentarité dans la modélisation et l'analyse des données.
* L'utilisation combinée du LIDAR, de la caméra et du ArUco code a permis une localisation robuste et fiable, essentielle pour le contrôle en temps réel.

Ces résultats valident la faisabilité de la prochaine étape du projet : le passage au prototypage physique, qui permettra de tester les mêmes concepts dans un environnement réel.