

# *Beach Poly’Bot*

# Introduction

Dans le cadre de notre projet de robotique de l’année 2022-2023, nous avons décidé après nous être rendu sur les plages non loin de notre école, de nous lancer le défi de nettoyer ces dernières.

A cause de la surutilisation de plastique dans l’industrie et la négligence de certains usagers, les différents littoraux se sont retrouvés remplis de déchets plastiques, de mégots de cigarettes et encore de nombreux autres détritus qui ne devraient pas se retrouver sur les plages.

Nous allons donc devoir créer un robot capable de se mouvoir dans le sable, sur des pentes faibles, qui serait capable de ramasser des déchets en surface et légèrement enfouis sous le sable, puis de faire un tri entre les réels détritus et les coquillages ou autres matériaux inertes. Il évoluera dans une zone sèche délimitée, avec une base où il pourra se recharger et déposer sa récolte, l’objectif étant de le faire fonctionner de nuit quand les plages ne sont pas occupées.

Le robot devra être entièrement autonome afin de réaliser sa tâche sans besoin d’assistance.

Nous avons découpé le projet en blocs fonctionnels qui interagissent entre eux de la façon décrite sur la figure ci-dessous.

**description du projet, un état d'avancement et ce qu'il reste à faire.**

### Compte rendu de mi-projet

---

#### 1. Implémentation du LIDAR

L'intégration du capteur LIDAR, modèle RPLIDAR A1, sur le robot permet une cartographie en temps réel de l'environnement, indispensable pour la navigation autonome et l'évitement d’obstacles. \*\*Connexion physique\*\* : Le capteur a été connecté au Raspberry Pi 3B+ via les broches GPIO, permettant une communication série UART avec le microcontrôleur. Une attention particulière a été portée à l'alimentation, le LIDAR nécessitant une alimentation stable de 5V pour un fonctionnement optimal. \*\*Configuration logicielle\*\* : L’installation du SDK de RPLIDAR a permis de faciliter l’acquisition des données en polar. Les premiers tests ont confirmé une détection d’obstacles jusqu'à 12 mètres avec une résolution angulaire précise. \*\*Développement d’algorithmes de traitement\*\* : Les informations de distance sont ensuite traitées pour créer une carte 2D de l’environnement du robot. Cette carte est mise à jour en continu pour permettre une navigation réactive. Les prochaines étapes incluent l'optimisation du temps de traitement des données et l'amélioration de la précision des relevés pour les petits obstacles.



---

#### 2. Implémentation du Filtre de Kalman sur la Position du Robot

L'implémentation du filtre de Kalman vise à améliorer la précision de la localisation du robot en fusionnant les données du GPS et de l’accéléromètre. \*\*Principe de fusion de données\*\* : Le GPS fournit une position approximative avec une marge d'erreur de 2 mètres, tandis que l’accéléromètre capture les accélérations linéaires du robot. En utilisant le filtre de Kalman, les mesures GPS sont corrigées en temps réel avec les informations de vitesse et d’accélération, permettant de réduire l'incertitude de la localisation. \*\*Étapes d’implémentation\*\* : Les étapes du filtre de Kalman (prédiction et mise à jour) ont été codées en Python pour s’exécuter en parallèle des autres processus. La position estimée est mise à jour à chaque cycle de mesure en utilisant les dernières données d’accélération et de position GPS. \*\*Résultats intermédiaires\*\* : Les tests ont montré une amélioration significative de la précision, l'erreur de position étant réduite de près de 50 % par rapport aux données GPS brutes. Les prochaines améliorations viseront à affiner le modèle de mouvement du robot pour encore améliorer la localisation.

Une image contenant texte, ligne, Tracé, Police

Description générée automatiquement

---

#### 3. Implémentation de la Caméra sur le Robot

La caméra intégrée est utilisée pour la reconnaissance d'objets et la détection de cibles, jouant un rôle essentiel dans la surveillance autonome. \*\*Configuration matérielle\*\* : La caméra, connectée au port CSI du Raspberry Pi, est configurée pour capturer des images en haute résolution. \*\*Traitement d’image\*\* : L'algorithme de traitement d’image est basé sur OpenCV et utilise des méthodes de détection de contours et de segmentation de couleur pour isoler des objets spécifiques dans le champ de vision. \*\*Mise en œuvre de la détection de cibles\*\* : Un modèle de reconnaissance a été entraîné pour identifier des objets particuliers grâce à une bibliothèque d’apprentissage automatique, augmentant la capacité du robot à repérer des cibles prédéfinies. \*\*Premiers résultats\*\* : Les essais ont permis une détection correcte des objets dans des environnements variés, mais des améliorations sont prévues pour la reconnaissance en conditions de faible luminosité et pour le suivi de cibles mobiles.

---

Ce bilan intermédiaire met en avant les progrès réalisés et identifie les axes d’amélioration pour optimiser chaque composant du système.

# Conclusion

Ainsi, au vue des différentes possibilités que nous avons exploré précédemment, notre choix s’est donc porté sur un système de ramassage par herse, remontée à l’aide d’un ou de deux vérins, déposant les objets ramassés sur un convoyeur. Ce convoyeur faisant passer les objets devant une caméra de reconnaissance afin de permettre au bras de tri orienté à l’aide d’un servo moteur de rejeter ou de stocker l’objet. Le robot se déplacera à l’aide de quatres roues motrices mises en mouvement par des moteurs à courant continu et orienté par des moteurs pas à pas.

# Bibliographie

Références bbliographiques :

[Motion control tips]

<https://www.motioncontroltips.com/selecting-the-right-batteries-and-motors-for-battery-powered-commercial-equipment/>

**Un compte-rendu (2 pages MAX) de mi-projet devra nous être remis au plus tard le mardi 5 novembre : il devra comporter une description du projet, un état d'avancement et ce qu'il reste à faire.**

**Vous m'enverrez votre rapport au format PDF par mail.**