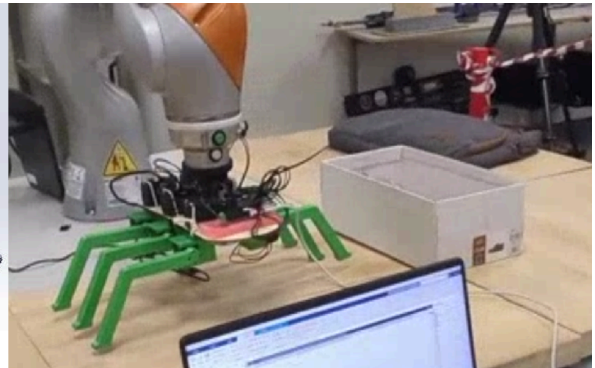
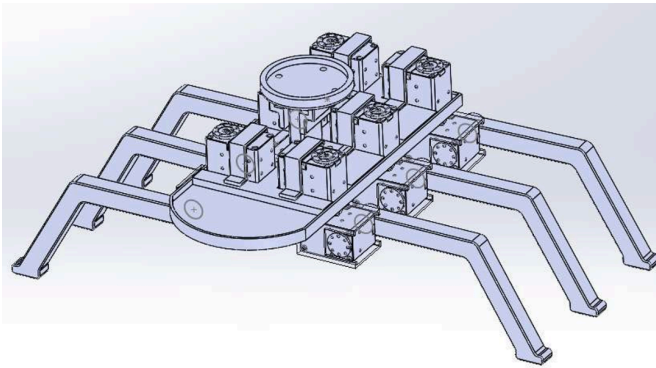

CONCEPTION ROBOTIQUE

SPIDER BOT



Groupe CPT

Arnaud GODET, Angel BABOUHOT, Gabriel SANCHEZ DE LEON, Nicolas HERAULT, Mathias FOYER

SOMMAIRE

INTRODUCTION	3
I - Cahier des charges	4
I-1. Contexte et objectifs	4
I-2. Fonctionnalités attendues	5
I-3. Livrables	5
I-4. Défis et contraintes	5
I-5. Nombre d'heures alloué au projet :	5
II - Matériel à disposition	6
III - Inspirations	7
IV - CAO	8
V - Contrôle matlab	10
VI - Contrôle moteurs	12
VII - Difficultés	13
VIII - Pistes d'amélioration	14
CONCLUSION	16

INTRODUCTION

Le projet SpiderToKukaHand est une initiative combinant robotique, conception mécanique et programmation. Son objectif principal était de transformer une main robotique de Kuka en un robot autonome fonctionnant comme une araignée. Ce projet, réalisé dans le cadre des études à l'école d'ingénieurs Junia HEI Châteauroux, a impliqué des compétences interdisciplinaires pour concevoir, contrôler et intégrer les mécanismes robotiques nécessaires à cette transformation.

La réalisation s'appuie sur l'utilisation du KST Kuka Sunrise Toolbox pour le contrôle du robot Kuka LBR7 et sur des composants tels que les cartes OpenRB et les moteurs Dynamixel pour la main araignée. Ce projet a permis aux étudiants de développer des compétences techniques tout en expérimentant les défis de la robotique collaborative et autonome.

I - Cahier des charges

I-1. Contexte et objectifs

Le projet SpiderToKukaHand vise à concevoir une pince robotique innovante capable de se fixer à l'extrémité d'un bras robotique Kuka pour réaliser des opérations de pick-and-place, tout en ayant la capacité de se détacher et de se déplacer de manière autonome sur le modèle d'un robot araignée. Ce projet, réalisé dans le cadre d'un cursus d'ingénierie à Junia HEI Châteauroux, exploite les synergies entre robotique, conception mécanique et programmation avancée.

Objectifs du projet :

1. **Conception d'une pince multifonctionnelle** : Créer une pince robuste et performante adaptée aux tâches de manipulation via le bras Kuka.
2. **Modularité et autonomie** : Offrir une transition fluide entre un mode "fixé au bras" pour les tâches de pick-and-place et un mode "robot autonome" pour explorer ou interagir avec son environnement de manière mobile et agile.
3. **Intégration et collaboration** : Garantir une interaction fluide entre les différents modules mécaniques et les systèmes électroniques, tout en assurant un contrôle précis et réactif dans les deux configurations.

Ce projet met en avant les défis de modularité, de conception biomimétique, et d'autonomie dans un contexte où polyvalence et innovation sont des priorités.

I-2. Fonctionnalités attendues

- Détachement de la main du robot Kuka en un robot araignée fonctionnel.
- Contrôle précis et coordonné des moteurs pour une prise d'objets efficace.
- Contrôle précis et coordonné des moteurs pour une mobilité réaliste.

I-3. Livrables

- Une main robotique transformée en robot araignée, capable de réaliser des déplacements autonomes.
- Présentation orale des travaux effectués.
- Vidéos ou démonstrations pratiques mettant en avant les performances du système développé.
- Documentation détaillant les choix de conception, les étapes de réalisation et les résultats obtenus.

I-4. Défis et contraintes

- Intégration efficace entre la mécanique, l'électronique et la programmation.
- Optimisation des performances des moteurs Dynamixel et des cartes OpenRB.

I-5. Nombre d'heures alloué au projet :

- 6,5 demies journées = 48h et 45 min
- 5 soirs environ 3h / soir = 15h.

Ce cahier des charges sert de guide structurant pour la conception et la réalisation du projet **SpiderToKukaHand**.

II - Matériel à disposition

Le matériel mis à notre disposition pour ce projet est le suivant :

- Robot Kuka LBR7 du hall mécatronique



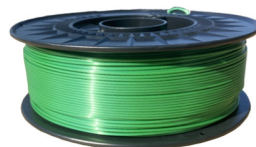
- Carte électronique OpenRB-150



- 12 servomoteurs Dynamixel XL430-W250



- Une bobine de PLA pour impression 3D



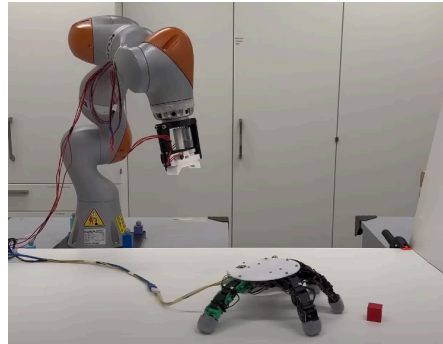
- Accès ou différents outils du FabLab de la CCI

III - Inspirations

- Travail des chercheurs à l'EPFL : <https://www.youtube.com/watch?v=3xBnIZkcll8>



Mode 1 : Pince

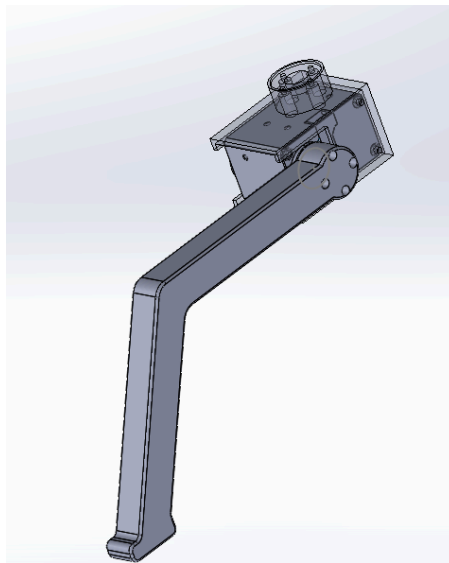


Mode 2 : Déplacement

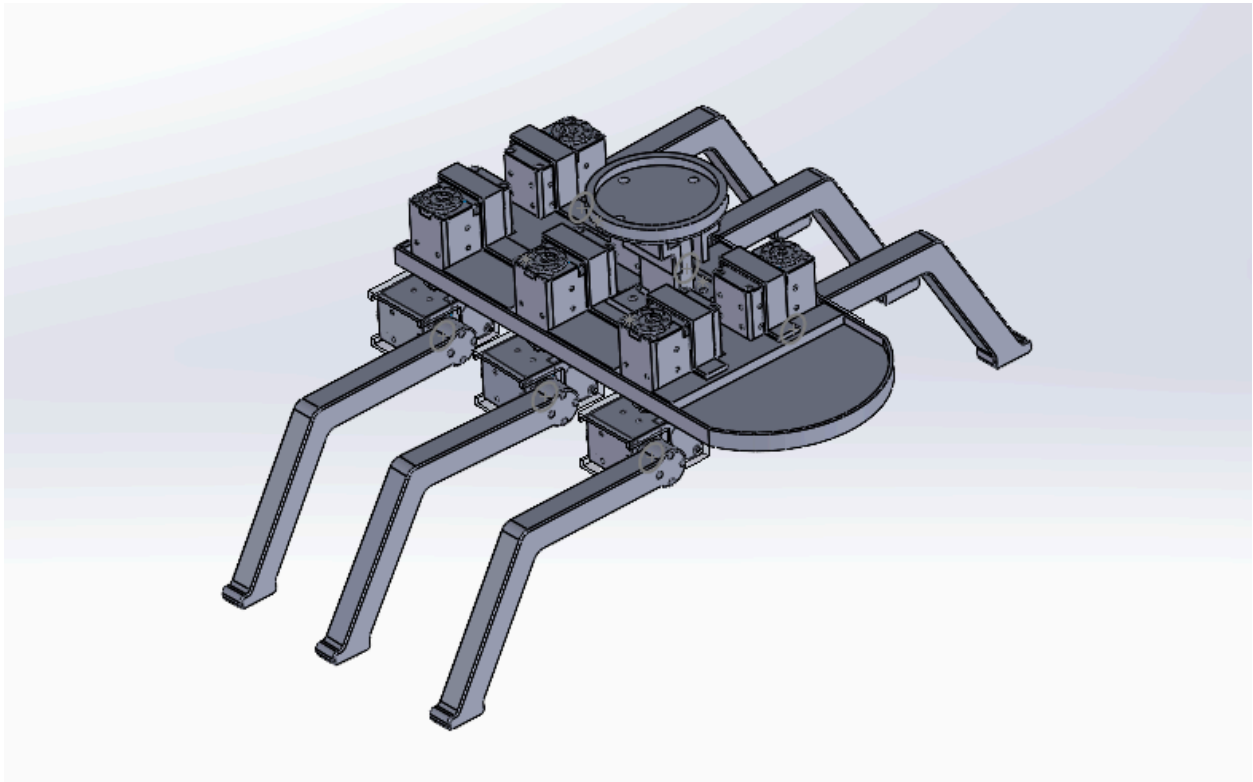
- Travaux de l'ISM sur AntBot :
<https://www.defense.gouv.fr/aid/actualites/antbot-biomimetisme-au-service-navigation-celeste>

IV - CAO

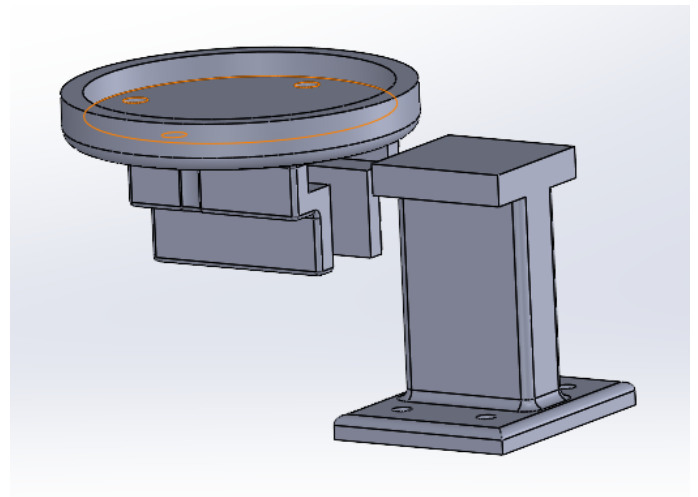
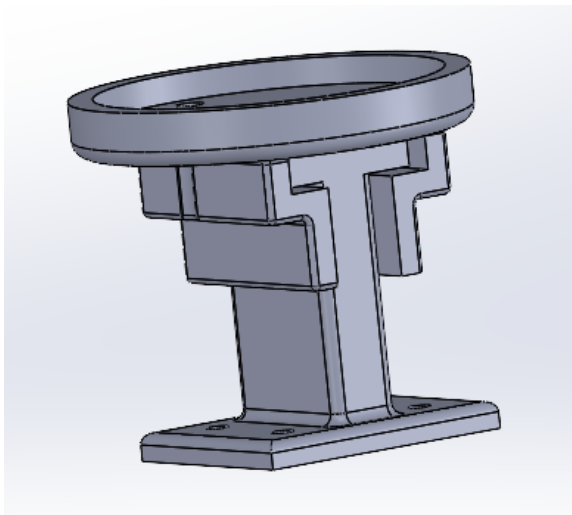
Grâce aux inspirations précédemment citées nous avons conçu un robot composé de 6 pattes afin de garantir la stabilité par le fait d'avoir en permanence 3 points d'appui. Chaque patte est composée de 2 moteurs, un moteur pour le déplacement horizontal et un pour le déplacement vertical. De plus, nous avons fait en sorte de ne pas mettre les pattes exactement l'une en face de l'autre pour permettre d'attraper les objets en mode pince.



Le tout est réuni grâce à une plaque centrale sur laquelle nous fixons les moteurs à déplacement horizontal et la carte de contrôle. Les moteurs de chaque patte sont assemblés grâce à une pièce de liaison.

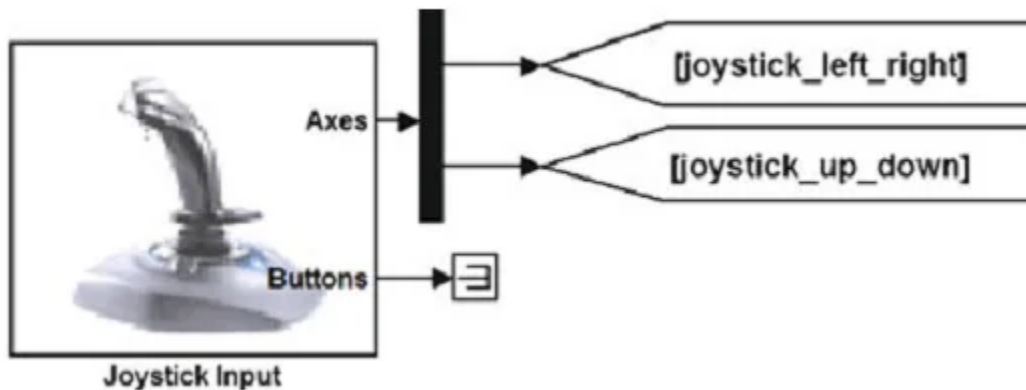


Pour ce qui de la partie permettant la liaison entre le robot et le kuka nous avons décidé de réaliser une liaison glissière permettant le détachement par un simple déplacement du kuka.



V - Contrôle matlab

Dans un premier temps nous avons expérimenté avec l'objet joystick input dans Simulink pour récupérer les inputs de notre manette xbox et manipuler une simulation matlab avec dans laquelle on déplaçait un cercle dans un espace 2D.



Ensuite sous matlab nous avons utilisé l'objet vrjoystick(ID) et la librairie KST pour contrôler le robot kuka en lui ordonnant de suivre la position de notre sphère simulée dans un espace 3D dont il devait suivre les coordonnées avec un offset et la fonction "movePTPJointSpace". Ce programme fonctionnait mais les déplacements n'étaient pas fluides car le robot s'arrêtait au passage de chaque point.

Après avoir testé et amélioré les déplacements du robot en utilisant le mode **startDaDirectServo**, nous avons développé un programme plus avancé pour intégrer un contrôle en temps réel basé sur les entrées d'un joystick. Ce programme permet de piloter le bras robotisé Kuka de manière fluide et sécurisée tout en respectant les contraintes mécaniques du robot.

Initialisation :

Le programme commence par configurer la connexion avec le robot Kuka et déplacer celui-ci vers une position initiale définie en radians. Une transformation initiale est calculée à l'aide de la cinématique directe (Direct Kinematics).

Lecture des commandes utilisateur :

Les axes du joystick sont filtrés pour éliminer les zones mortes (deadzones) et garantir une commande précise. Les boutons du joystick permettent également de modifier les déplacements verticaux du robot.

Calcul des mouvements :

Les entrées du joystick sont utilisées pour générer des commandes de vitesse linéaire et angulaire. Ces commandes sont appliquées à une matrice de transformation représentant la position et l'orientation de l'outil dans l'espace 3D.

Résolution de la cinématique inverse :

Les nouvelles positions des articulations sont calculées via un solveur de cinématique inverse (Inverse Kinematics Solver) pour ajuster le robot en fonction de la matrice de transformation mise à jour.

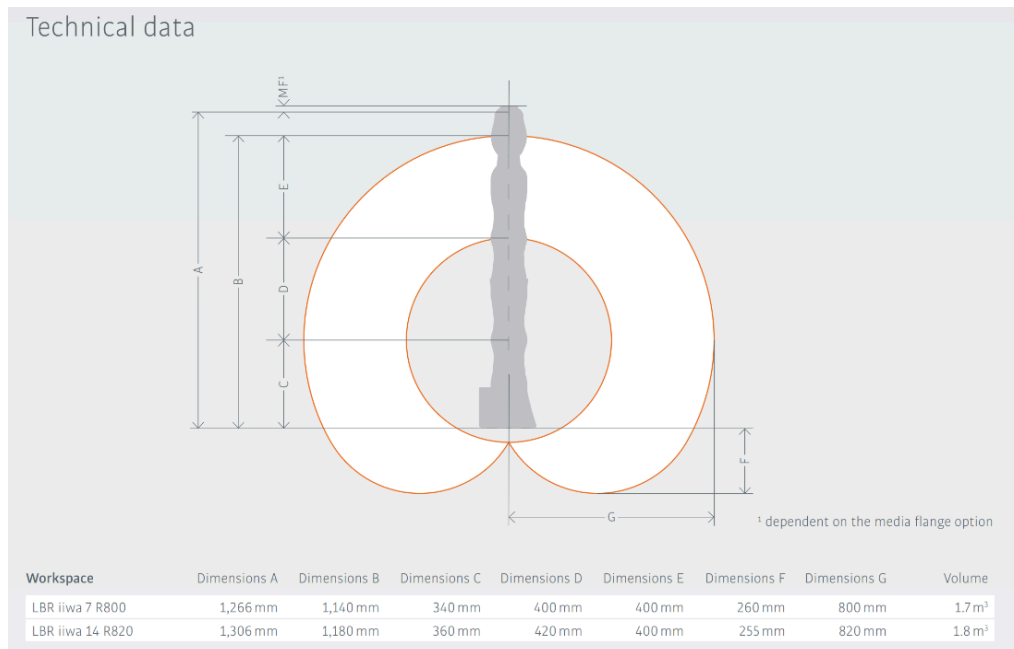
Validation et contrôle des limites :

Afin d'assurer la sécurité des mouvements, des vérifications sont effectuées sur les écarts entre les positions précédentes et les nouvelles positions calculées. Si un écart dépasse un seuil défini, les commandes sont annulées pour éviter des erreurs ou des dommages mécaniques.

Envoi des commandes au robot :

Les positions validées des articulations sont envoyées au robot pour exécution en temps réel, permettant une réponse rapide et fluide aux commandes utilisateur.

Workspace du Kuka :



L'espace de travail de notre système en mode pince est similaire à celui du robot Kuka puisque notre robot se greffe en tant qu'organe terminal de ce dernier.

Le programme final inclut à la fois ce contrôle du robot kuka et lorsque l'on appuie sur la touche "start" de la manette on change de mode pour communiquer via un port série avec la carte du spiderBot et lui envoyer des actions à effectuer. On peut ainsi lui demander de marcher, se fermer ou s'ouvrir. Si on appuie sur "select" on relance le programme pour revenir en mode contrôle du robot Kuka, on peut ainsi alterner d'un mode à l'autre pour déplacer le robot Kuka ou le spiderBot.

VI - Contrôle moteurs

Pour pouvoir alimenter les moteurs, nous les branchons sur la carte électronique OpenRB-150 disposant de 4 ports. Afin de limiter la charge sur les ports nous avons réparti équitablement les moteurs à hauteur de trois moteurs par port. Le contrôle de ceux-ci se fait via la bibliothèque Dynamixel2Arduino.h en utilisant la fonction `dxl.setGoalPosition(id, goal_position_deg[i], UNIT_DEGREE)`.

Dans le but de simplifier l'utilisation de cette commande nous passons par un modèle géométrique inverse. Le pilotage du robot en mode araignée se modélise comme ceci:



Actuellement, le MGI nous permet de prendre en compte uniquement les coordonnées X et Z car nous ne disposons que de deux moteurs et ainsi deux degrés de liberté.

Le pilotage du robot en mode main se fait directement via les angles en position ouverte et fermée:

```
graph LR; A[Pilotage direct en degrés];
```

Dans le but de protéger le matériel, une fonction regarde si l'angle donné au moteur est bien compris dans un intervalle prédéfini pour chaque mode.

VII - Difficultés

Au cours du développement du projet **SpiderToKukaHand**, plusieurs défis techniques et conceptuels ont été identifiés :

1. Masse élevée du SpiderBot

Le poids total du robot araignée s'est avéré relativement élevé, ce qui a occasionné une surcharge des moteurs lors des déplacements. Ce problème a été aggravé par la faible tolérance dans la position des pattes, rendant les mouvements contraignants pour les moteurs. Une augmentation des tolérances des positions a permis de limiter ces surcharges et d'éviter les erreurs des moteurs.

2. Définition des coordonnées et repères des moteurs

La gestion des coordonnées propres à chaque moteur a posé des difficultés, en particulier pour assurer une synchronisation et une précision optimales des mouvements. Chaque moteur nécessite un repère spécifique et un positionnement rigoureusement défini pour que les pattes fonctionnent de manière coordonnée. L'établissement de ces repères a nécessité un travail approfondi de modélisation et d'expérimentation, ce qui a complexifié le développement du contrôle du robot.

3. Singularité et limites des mouvements

Le bras Kuka peut entrer en erreur ou rencontrer des singularités lorsqu'il effectue des mouvements brusques ou lorsque les articulations s'écartent fortement de leur position initiale. Ces situations peuvent compromettre la sécurité et la précision des opérations. Pour prévenir ces problèmes, une limite dynamique a été intégrée dans le code. Ce mécanisme surveille en temps réel les écarts des positions des articulations et compare ces écarts à un seuil défini. Si une déviation importante est détectée, le système réagit en annulant le mouvement problématique et en revenant à une configuration sûre. Ce dispositif a permis de réduire les risques d'erreurs et d'améliorer la fiabilité des mouvements du bras robotisé.

VIII - Pistes d'amélioration

Pour améliorer la conception et les performances du projet **SpiderToKukaHand**, plusieurs axes d'amélioration ont été identifiés au cours de sa réalisation :

1. Ajout d'un troisième moteur par patte

Actuellement, chaque patte de l'araignée dispose de deux moteurs, ce qui limite ses mouvements à des arcs de cercle pour avancer. L'ajout d'un troisième moteur permettrait un contrôle complet en **x, y et z**, offrant ainsi une meilleure précision dans les déplacements et une marche plus fluide.

2. Passage à un design à cinq pattes

Le prototype actuel comprend six pattes, optimisées pour la stabilité. Cependant, pour obtenir un effet visuel plus proche d'une main, le robot pourrait être repensé avec cinq pattes. Ce choix refléterait mieux la transition entre la main Kuka et le robot araignée. Faute de temps et de disponibilité d'imprimantes 3D, nous n'avons pas pu explorer cette option ni construire plusieurs itérations.

3. Utilisation d'un électroaimant pour l'accroche

Le système d'accroche actuel repose sur une interface mécanique avec une partie mâle s'insérant dans une partie femelle. Bien que fonctionnel, ce système nécessite une grande précision et une bonne adhérence au sol lors de la séparation du bras Kuka. L'usage d'un **électroaimant** aurait simplifié ce processus, en permettant une activation et une désactivation précises pour soulever et détacher le robot. Cependant, nous ne disposons pas d'électroaimants suffisamment puissants pour réaliser ce concept.

4. **Structure plus rigide**

Par manque de ressources, seules les pattes et les supports moteurs ont été imprimés en 3D, tandis que le corps du robot était constitué d'une plaque de bois découpée à la découpeuse laser. Ce choix a rendu la structure moins rigide, avec une tendance à plier ou à se casser sous des contraintes importantes. Une refonte complète avec des matériaux plus robustes serait nécessaire pour améliorer la solidité du robot.

5. **Réduction de la taille de la structure**

La taille actuelle du robot a été définie en fonction des moteurs disponibles, qui imposent des dimensions assez larges. Avec des moteurs plus compacts, il serait possible de réduire les dimensions du robot, rendant sa conception plus optimisée pour ses objectifs.

6. **Amélioration de la marche**

Bien que fonctionnelle, la marche actuelle manque de fluidité. Une optimisation des algorithmes de contrôle des moteurs et une coordination plus précise entre les pattes permettraient d'obtenir un déplacement plus naturel et efficace.

7. **Passage sur batterie**

La carte qui contrôle les moteurs est actuellement alimentée sur secteur. Une amélioration possible serait d'intégrer une batterie permettant ainsi d'agrandir l'espace de travail en mode araignée actuellement limité par la longueur du câble d'alimentation.

CONCLUSION

Le projet **SpiderToKukaHand** a permis de répondre aux objectifs fixés dans le cahier des charges. Nous avons conçu et réalisé un système fonctionnel combinant le contrôle d'un bras robotique Kuka et la transformation de sa pince en un robot autonome inspiré du modèle araignée. Les démonstrations réalisées montrent que le prototype répond aux attentes initiales en termes de modularité, de mobilité et de performances techniques.

Cependant, des pistes d'améliorations ont été identifiées. Avec plus de temps et de ressources, nous aurions pu :

- modifier le design du robot pour qu'il ait 5 pattes et se rapprocher d'une main
- Intégrer un troisième moteur par patte pour un contrôle plus précis en 3D.
- Tester un système d'accroche basé sur un électroaimant pour simplifier le détachement du robot.

Ce projet a été une opportunité unique de mettre en pratique nos compétences en robotique, mécanique et programmation, tout en collaborant pour surmonter des défis complexes. Nous sommes fiers du chemin parcouru et des enseignements tirés, qui pourront guider les prochaines évolutions de ce prototype.