

SORBONNE UNIVERSITÉ

STAGE EN LABORATOIRE

---

# Exploration morphologique et évaluation de module IMU pour des robots centimétriques

---

*Auteurs :*

Matej MAYET

Valentin DUVIVIER

*Superviseur :*

Pr. Nicolas BREDECHE

9 décembre 2019



## Remerciements

Nous profitons par le biais de ce rapport pour adresser quelques mots de remerciements à notre tuteur de stage le Professeur Nicolas Bredeche, pour son aide précieuse et la diversité de ses conseils, tant sur le plan technique que sur le plan méthodologie de travail de notre stage. Nous avons beaucoup appris concernant la rigueur qu'il faut avoir sur ce type de projets à long termes, et cela nous sera forcément utile pour de futurs stages ou projets de ce type.

Par ailleurs une aide certaine nous a été fournie par des collaborateurs à l'ISIR de ce dernier, notamment Fabien Caura et Nicolas Fonbonne qui nous ont aidés en grande majorité, et débloqués à maintes reprises concernant la partie IMU.

## Résumés

### En français

Actuellement, une grande branche de la robotique est associée à l'étude de certains comportements, et ceci via des algorithmes. On développe alors des algorithmes reproduisant ces comportement dans le but de les modéliser numériquement et de les étudier avec précision. Ce type d'étude se base notamment sur des observations sur les animaux comme les fourmis, où l'on tente de reproduire numériquement leur comportement de travail de groupe, on appelle cela plus communément la coordination et communication en essaim [13]. Une fois cette démarche numérique effectuée, l'une des phases importantes sera alors de vérifier le fonctionnement des algorithmes et leur application concrète. Il existe différentes manières de les expérimenter : on peut le faire de manière virtuelle grâce à une simulation 2D ou 3D (selon la nature de l'algorithme qu'on utilise) [2], ou bien on peut directement observer le comportement réel des robots munis de l'algorithme en question. Dans ce dernier cas, des robots biens réels avec une masse, et auxquels les lois physiques s'appliquent sont testés dans un cadre qui dépasse la simple modélisation virtuelle.

En privilégiant une étude réelle, on va s'éloigner du cadre numérique qui néglige certaines contraintes physiques, et ainsi mieux coller au cas concret.

Une contrainte qui se présente à cette étape est le moyen que l'on va mettre en place pour tester en condition réelle ces robots : en terme de place, de nombre de robots nécessaire, etc. Pour ce qui est des robots à déployer, il faut que ceux-ci soient peu cher, facilement reproductible, de petite taille et dont les capacités techniques soient en accord avec la fonction que leur alloue l'algorithme. De ce fait, nous arrivons à l'objet de notre stage : les Kilobots. Ce sont de petits robots capables d'effectuer des actions de déplacement coordonnées, créés par deux chercheurs à l'Université d'Harvard et qui sont parfaitement adaptés pour la simulation de systèmes de communication en essaim. [1]

En effet, leur capacité à recevoir des informations émises depuis une base de contrôle rend certains types de simulations particulièrement efficaces. Par ailleurs, ils répondent aux critères que nous venons d'énumérer à savoir les critères de prix, de taille, etc. Il y a cependant certains aspects du robot qui peuvent être modifiable dans le but de rendre son utilisation plus polyvalente.

C'est avec cela en tête que nous avons abordé les fonctions principales du cahier des charges de notre stage. Nous avons ainsi par bien des façons au

cours de ces derniers mois analysés les différentes manières dont on aurait pu améliorer ce robot (vitesse de déplacement, précision de trajectoire, etc).

## In English

Currently, a big part of robotics is associated to the study of certain behaviors, and this via algorithms. We then have to develop algorithms reproducing those behaviors in order to make numerical modelisation and to study them.

This type of study is based on observations made on living being like ants. We then use those observations to reproduce numerically the ants team work's behavior ; this is what we commonly call coordination and communication as a swarm [13]. Once this numerical approach has been made, a big step for us to go through will be to check for the good working of the algorithms and of their concrete application. There are different ways to experiment them : we can do this virtually via a 2D or 3D simulation (according to the nature of the algorithm we're using) [2], or we can do it directly by observing the real behavior of the robots carrying the algorithm we were talking about. In this last case of a real condition application, the robots with real weight, and for which the Newton's laws apply, are tested in a way that overpass the virtual simulation.

By privileging these concretes tests, we will extract our analysis from the numerical context which neglects some physical constraints, and we will thus stuck with reality.

A constraint that appears at this stage is the way that we will use into real condition's testing for our robots : in terms of both place and the necessary amount of robots. To make simulations with a big number of robots, they have to be cheap, they got to be easily reproducible, short and whose technical capacities are in agreement with the function that the algorithm allocates to them. There is the link with our project : the Kilobots. They are small robots capable of performing coordinated motion actions, that has been created by two researchers at Harvard University and that are ideally suited for simulating swarm communication systems.

Indeed, their ability to receive information from a stationary base, makes some types of simulations particularly effective. Besides, they fit with the criteria enunciated before that are the fact of having cheap and tiny robots. However there are some aspects of the robot that are modifiable in order to make its use more versatile.

This is by keeping that in head that we approached the principals functions of our internship. Thereby, throughout the last few months, we have in many ways analyzed the various ways in which we could improve this robot (speed of movement, accuracy of trajectory, etc).

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Présentation de l'ISIR</b>	<b>5</b>
2.1	La structure . . . . .	5
2.2	Domaines de recherches . . . . .	5
<b>3</b>	<b>Bibliographie détaillée</b>	<b>5</b>
3.1	État de l'art - la communication / coordination en essaim . . . . .	5
3.2	Résumé de la recherche documentaire . . . . .	6
3.3	Mots clés . . . . .	7
<b>4</b>	<b>Étude macroscopique des Kilobots</b>	<b>7</b>
4.1	Qu'est ce qu'un Kilobot ? . . . . .	7
4.2	Comment faire fonctionner un (des) Kilobot(s) ? . . . . .	9
4.3	Tests sur les Kilobots modifiés . . . . .	9
4.3.1	Tests des Kilobots modifiés, première génération . . . . .	9
4.3.2	Premiers bilans . . . . .	12
4.3.3	Finalisation de l'analyse . . . . .	13
<b>5</b>	<b>Création de nos propres exosquelettes et étude bien plus poussée de ceux-ci</b>	<b>14</b>
5.1	Présentation des différents exosquelettes . . . . .	15
5.1.1	L'exosquelette "Brosse à dent" . . . . .	15
5.1.2	L'exosquelette "Triangle penché" . . . . .	16
5.1.3	L'exosquelette "Courbé" . . . . .	17
5.1.4	L'exosquelette "Spider" . . . . .	18
5.2	Protocoles de tests des exosquelettes . . . . .	18
5.2.1	Protocole de calcul de vitesse de rotation . . . . .	19
5.2.2	Protocole de calcul de trajectoire et vitesse de déplacement . . . . .	19
5.3	Résultats de nos tests . . . . .	20
5.3.1	Graphes de trajectoires . . . . .	21
5.3.2	Confrontation des résultats . . . . .	26
5.4	Bilan motricité . . . . .	28
<b>6</b>	<b>Implémentation d'une Centrale d'inertie / Inertial Measurement Unit (IMU)</b>	<b>29</b>
6.1	Code de localisation . . . . .	29
6.1.1	Modélisation 3D en temps réel d'un Kilobot . . . . .	31
6.2	Comparaison Théorie/Pratique - Application au cas du Thymio . . . . .	32
6.2.1	Protocole de relevé d'angles . . . . .	32
6.3	Résultats . . . . .	33
6.4	Etude des sources d'erreur . . . . .	36
6.5	Pistes concernant la correction de trajectoire . . . . .	36
<b>7</b>	<b>Conclusion - Bilan Stage</b>	<b>37</b>

# 1 Introduction

De nombreuses recherches ont été effectuées afin d'utiliser les Kilobots dans différents domaines. On sait par exemple maintenant les faire s'organiser pour qu'ils s'agencent de sorte à former une figure prédéfinie (comme une étoile ou une lettre de l'alphabet) [15]. Cet aspect est lié au domaine du positionnement des robots, et c'est l'une des premières choses qui ont été expérimentées grâce aux robots en essaim. Cette faculté est en grande partie due à leur capacité d'auto-organisation qui rend ce type de tâche particulièrement aisée.

Une étude en lien avec cet aspect cartographie des robots qui a été faite concernant les Kilobots est celle du KiloGrid [6]. Pour faire simple le but est d'étudier, grâce à une plateforme munie de capteurs et de couples émetteurs/récepteurs infra-rouge, la position d'un Kilobot en mouvement sur cette plateforme, et de lui donner des instructions en fonction de sa position. Cela peut être très utile car jusqu'ici le transfert de données entre les Kilobots et la base de contrôle était unilatéral : c'était la base qui envoyait des données / instructions aux robots et ceux-ci agissaient en fonction de ce que commandait cette base. Or grâce à cette plateforme, quand un Kilobot passe sur une cellule munie d'un capteur, celle-ci va envoyer à la base l'information : "Le Kilobot n°X est sur la cellule n°X", ce qui va donner une information sur la position du Kilobot. Le KiloGrid va ensuite passer par une interface USB qui agira comme un intermédiaire pour effectuer une communication du Kilobot vers la base.

Et, depuis un peu plus d'un an, l'Institut des Systèmes Intelligents et de Robotique (ISIR) à Sorbonne Universités travaille sur un système de localisation de Kilobot, qui serait plus précis que le KiloGrid en terme de localisation, et surtout qui permettrait de faire de la correction de trajectoire lorsque le robot serait dévié du chemin qu'il aurait du prendre. Car pour l'instant les robots ne sont capables que d'activer des vibreurs à droite et à gauche avec une certaine intensité pour avancer, ils ne sont pas équipés de correcteurs de trajectoire.

C'est là que réside toute la fonction de notre stage. Les objectifs qui nous ont été donnés étaient très clairs : améliorer la stabilité, vitesse et précision des robots, et faire fonctionner un moyen de récupération de données d'une centrale inertielle [8] embarquée sur un Kilobot.

La première partie de ce rapport sera dédiée à la présentation du laboratoire, des équipes de recherche travaillant au sein de celui-ci, et des thématiques abordées par l'ISIR. Suivi par une recherche bibliographique détaillée sur la caractéristique principale des Kilobots qu'est la communication en essaim. La troisième partie traitera de l'explication des Kilobots, de la manière dont ils fonctionnent et de leurs caractéristiques techniques. Suite à cela nous aborderons une étude macroscopique des Kilobots qui se penchera surtout sur l'optimisation de la vitesse de déplacement et la stabilité de ceux-ci. Nous présenterons les résultats de nos tests sur les différentes améliorations que nous avons faites, et pour finir on consacrerait la dernière partie à la récupération de données de position par la centrale inertielle et des différents protocoles concernant l'IMU (centrale inertielle).

## 2 Présentation de l'ISIR

### 2.1 La structure

L'ISIR est un laboratoire de recherche qui rassemble des chercheurs et enseignants-chercheurs dont le corps de métier relève de différentes disciplines des Sciences de l'Ingénieur et de l'Information ainsi que des Sciences du Vivant. Selon le site officiel de l'ISIR : "*L'ISIR est une Unité Mixte de Recherche (UMR7222) commune à l'Université Pierre et Marie Curie (UPMC) et au Centre National de la Recherche Scientifique (CNRS). L'ISIR est rattaché d'une part à la faculté d'Ingénierie de l'UPMC (UFR 919) et d'autre part à l'Institut des Sciences de l'Information et de leurs Interactions (INS2I) du CNRS. L'Institut national de la santé et de la recherche médicale (INSERM) est également tutelle de l'une de ses équipes, l'Equipe de recherche labellisée (ERL) U1150.*" [11].

Pour notre part nous avons fait notre stage au sein du laboratoire répondant à la dénomination "AMAC" (Architectures and Models for Adaptation and Cognition). Grossièrement, ce groupe de recherche met l'accent sur des recherches sur la mécanique des systèmes biologiques et sur la conception d'architectures servant au contrôle et à l'optimisation d'interactions inter-robots intelligents. [10]

### 2.2 Domaines de recherches

Pour ce qui est d'une description plus spécifique des axes de recherches menés par ce laboratoire, en font partie la modélisation et l'analyse des systèmes dynamiques artificiels et naturels ainsi que la conception optimale de systèmes robotiques interactifs. A cela on peut ajouter la commande des systèmes interactifs et la conception et le traitement du signal de systèmes perceptifs multimodaux. Et plus généralement la modélisation des interactions homme-système et l'apprentissage artificiel [10].

Mais ce qui est intéressant ce sont les applications auxquelles s'adressent tout ces concepts qui peuvent sembler un peu compliqués au premier regard. On peut citer beaucoup d'applications tournant autour de l'assistance au niveau médical, tels que l'assistance aux gestes chirurgicaux et à la rééducation fonctionnelle et enfin les interactions multimodales pour la robotique personnelle et le handicap. A cela s'ajoute une dimension qui s'adresse purement à la robotique, pour n'en citer que quelques uns, on a la robotique évolutionniste, la simulation et la synthèse de fonctions motrices pour la robotique humanoïde et les mannequins virtuels ou encore la perception visuo/acoustique active et 3D (pour cette dernière, cela peut également s'appliquer dans le domaine médical) [11].

## 3 Bibliographie détaillée

### 3.1 État de l'art - la communication / coordination en essaim

La robotique en essaim est une branche de la robotique appliquant les méthodes d'intelligence distribuée aux systèmes à plusieurs robots. Il s'agit généralement d'utiliser des robots simples, peu coûteux, facilement reproductible, d'un intérêt individuel assez limité, mais qui ensemble forment un système plus ou moins complexe qui

fonctionne comme une unité. La robotique en essaim cherche à étudier la conception et le comportement de tels robots. Une composante clé de l'essaim est la communication entre ses membres. L'avantage d'une telle communication est de pouvoir reconstituer, grâce à des algorithmes spécifiques, certains comportements animaliers, cellulaires, etc. Les difficultés inhérentes au domaine de la robotique en essaim sont la transmission de données fiable et la standardisation des modules intelligents. Les recherches portent donc aussi bien sur les aspects software et interactions entre les robots, que sur les aspects physique et matériel de ces derniers.

Dans ce sens, des travaux concernant la communication et l'optimisation matériel de plusieurs robots en essaim ont été lancés et ont aboutis pour certains à de grandes avancées scientifiques, faisant d'eux des acteurs majeurs dans leur domaine. Pour n'en citer que quelques uns : le projet SMAVNET de l'EPFL utilisant des drones volants pour reconstruire un réseau de communication [12], le projet sub-CULTron qui s'est terminé en novembre 2019 et qui utilisait une flotte de drones marins pour faire du monitoring de niveau d'oxygénation dans le lagon de Venise [3], ou encore le projet PHOENIX se servant d'un réseau de senseurs (mobiles mais pas actifs) autrement dit des robots portés par le courant marin ou aérien, ou par un flux d'eau (comme des canalisations par exemple) [4].

Notre stage s'inscrit plus particulièrement dans le domaine des interactions inter-robots et de l'optimisation de motricité pour des modules 2D. Il est centré autour d'un module spécifique particulièrement adapté à ces utilisations : le Kilobot [17]. Ce robot permettant une communication avec d'autres unités via des signaux infrarouges est disponible en grand nombre, à moindre coût et est facilement améliorable/modifiable. Notre rôle dans un projet doré et déjà lancé, est d'améliorer certaines caractéristiques morphologiques des kilobots déjà existants, dans le but de les rendre plus performants. Nous travaillerons aussi sur l'aspect communications en développant une recherche inédite sur l'évaluation de module IMU, servant à la cartographie de robot et à la correction de trajectoire.

## 3.2 Résumé de la recherche documentaire

La toute première chose que nous avons fait est de lire la documentation fournie par notre encadrant Nicolas Bredeche. Il a pu nous fournir des documents précieux rédigés par des chercheurs membre de l'ISIR ainsi que des documents d'anciens Master qui ont travaillé auparavant sur le même projet, avec un objectif toutefois différent du nôtre (sans quoi notre travail n'aurait aucune utilité).

Aussi nous avons pu chercher de notre côté, grâce à certains outils tels que Term-Sciences, Techniques de l'ingénieur ou encore le Grand dictionnaire terminologique, des mots clés et des pistes de recherches pour débiter notre stage. Nous avons ainsi mis la main sur divers documents constructeurs et des travaux sourçant les équipes qui ont créé le Kilobots, à Harvard. C'est une partie que nous avons tendance à oublier, mais la compréhension du sujet et de l'environnement dans lequel nous allons baigner pendant quelques mois est important afin d'être le plus efficace possible dans nos recherches.

Par la suite, nos connaissances sur le sujet étant plus affinées et précises, nous avons commencé à nous documenter sur des aspects plus spécifiques de notre stage, en utilisant notamment Google Scholar. En fait, une fois que notre sujet était clair

dans notre esprit, il a fallu que l'on s'acclimate aux logiciels que nous devrions utiliser pour l'étude de nos robots. Par exemple, un des aspects primordiale à ce moment ci était de prendre en main l'environnement Linux qui nous était assez peu familier, pour ce qui est de la compilation des programmes, l'extraction de fichiers compressés, l'installation de logiciels de modélisation 3D etc. Cet outil, qui est un équivalent de Windows, nous était alors inconnu et il a fallu faire une phase de recherche sur ce point dans le but de faciliter nos échanges avec le Kilobot pour nos études à venir.

Une remarque reste cependant pertinente, la majorité des documents que notre encadrant nous a envoyé sont des documents qui, malgré le fait qu'ils nous aient beaucoup aidés, ne sont ni publiés sur des sites d'articles scientifiques ni suffisamment sourcés. Ainsi ils ne peuvent pas figurer dans la bibliographie, malgré l'aide sérieuse qu'ils nous fournis au cours du stage. Il a donc fallu, à partir de nos recherches, revenir aux documents primaires, c'est à dire ceux étant issus de recherches scientifiques et présentant une documentation fournie et détaillée.

Pour finir, on peut également faire référence à des sites de chercheurs et / ou de passionnés d'informatique qui expliquent via des tutoriels comment utiliser certains logiciels tels que KiloGui pour téléverser des programme de l'ordinateur jusqu'à un Kilobot, ou encore *Processing IDE* qui nous a servi à modéliser un Kilobot en 3D, donnant une compréhension visuelle de nos problématiques numériques.

### 3.3 Mots clés

Kilobot, Asymetrical, swarm, swarm robotics, coordination in multi-robot systems, IMU (Inertial Measurement Unit), gyroscope, accelerometer, MEMS (Microelectromechanical systems) fusion, Holonomic basis, Kilogrid, IR communication, collision detection, noise avoidance, motion capture.

## 4 Étude macroscopique des Kilobots

### 4.1 Qu'est ce qu'un Kilobot ?

Un Kilobot est un petit robot de 33mm de diamètre capable de se déplacer grâce à la force de vibration de ses deux moteurs. Grâce à la disposition asymétrique de ses trois pattes et au positionnement des deux vibreurs de part et d'autre de la patte avant, il va pouvoir effectuer des déplacement en fonction de l'intensité des vibrations des moteurs.

Si un moteur vibre et l'autre non, il va tourner sur place dans le sens opposé au côté du vibreur qui vibre (si le vibreur à droite de la patte avant est actif, il va tourner à gauche, et inversement) ; si les deux moteurs vibrent à la même intensité le robot avance tout droit ; si un moteur vibre plus avec plus d'intensité que l'autre, le Kilobot avance en tournant sur lui même (son sens de rotation étant déterminé de la même manière que pour une rotation simple). On note que plus l'écart entre l'intensité des deux vibreurs est grande, plus le robot penchera vers une rotation plutôt que vers une translation ; si aucun moteur n'est actif le robot n'effectuera évidemment aucune action. Il est d'ailleurs incapable de reculer, il doit d'abord ef-



fectuer une rotation de 180 degrés pour ensuite avancer tout droit.

Pour actionner ces moteurs et alimenter les composants électroniques nécessaire à son fonctionnement, le Kilobot est équipé d'une pile rechargeable 3.7 Volts. Celle-ci est en contact direct avec un arc métallique qui s'accroche sur une borne de recharge, une longue barre métallique branchée sur une prise secteur qui va appliquer une différence de potentiel et recharger la pile. Par ailleurs cette borne de recharge facilite grandement le stockage des Kilobots pendant leur recharge.

Pour ce qui est du reste des composants électroniques, la plupart sont des composants liés à la communication de données. La LED par exemple sert d'interface de communication machine-homme, dans le but d'avertir l'utilisateur de l'état de charge du robot par exemple, ou encore d'avertir si le Kilobot est en phase de Boot-loading, c'est à dire si le robot est dans une phase où il est capable de recevoir des données envoyées par la base de contrôle [16]. Deux autres composants sont également ceux utilisés en tant qu'interface machine-machine, dans le but de recevoir et d'émettre des données Infra-Rouge : les émetteur et récepteur Infra-Rouge situés sur la face inférieure du Kilobot.

On peut retrouver sur la *Figure 1* ci-dessous, la position de chacun des composants mentionnés précédemment sur le Kilobot, ainsi que leur forme en 3D [19].

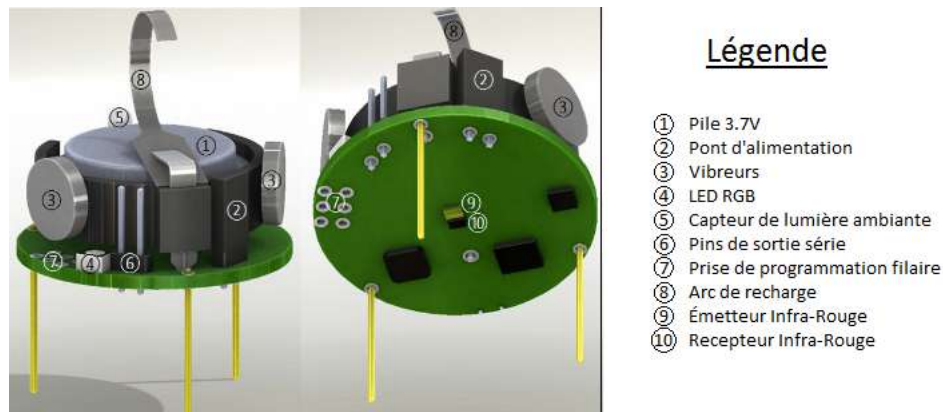


FIGURE 1 – Schéma et composants d'un Kilobot

## 4.2 Comment faire fonctionner un (des) Kilobot(s) ?

La première étape importante est de prendre un (des) Kilobot(s) chargé(s) et de le(s) placer en dessous de la base de contrôle sur une surface réfléchissante. En effet pour tout ce qui est du transfert de données, la base de contrôle envoie des signaux infra-rouge depuis une certaine hauteur au dessus des Kilobots, et étant donné que les récepteurs infra-rouge de ceux-ci sont situés sur leur face inférieure, la partie haute du robot gêne les signaux lors de la transmission, ainsi la surface réfléchissante va agir comme un miroir et les signaux vont "rebondir" sur celle-ci et arriver jusqu'au(x) récepteur(s) infra-rouge du (des) Kilobot(s) sans aucun problème.

La prochaine étape est de brancher la base de contrôle à l'ordinateur, et grâce au logiciel KiloGUI on va importer le programme qui va donner des instructions au(x) robot(s), ce fichier est en .hex car c'est l'extension de fichier que KiloGUI utilise pour le compiler dans le Kilobot.

En suivant bien les instructions des manuels déjà disponibles sur le fonctionnement de KiloGUI [18], on arrive à téléverser le programme dans la mémoire interne du (des) robot(s).

## 4.3 Tests sur les Kilobots modifiés

On entre à présent dans la première partie du travail concret de notre projet. Cette première partie concerne, dans un premier temps, le test complet des modifications et des exosquelettes déjà existants sur les Kilobots et, dans un second temps, le test des modifications et exosquelettes que nous avons créés nous même. Les deux parties sont distinctes et il est important de les différencier car la première partie se consacre exclusivement à une étude visuelle, où l'on va déterminer selon certains critères quelles sont les qualités et les défauts de chaque module, avant de les confronter et en déduire les formes et la matière avec lesquels nous allons concevoir nos propres prototypes. Ainsi la deuxième partie est consacrée à ces prototypes "hybrides", qui utilisent chacun des aspects des modules de la première génération (c'est comme cela que nous appellerons les exosquelettes créés par les membres de l'ISIR qui nous servent de base), pour faire des prototypes optimisés. Donc, on utilise les résultats de la première partie, pour la seconde. Une fois ces prototypes deuxième génération créés (c'est comme cela que nous appellerons les exosquelettes que nous avons créés), on fait des tests bien plus rigoureux concernant leurs caractéristiques et on détermine quel prototype est le meilleur et on émet des hypothèses sur le domaine où son utilisation serait la plus adaptée.

### 4.3.1 Tests des Kilobots modifiés, première génération

Pour cette phase de tests on va s'intéresser à 8 prototypes de Kilobots. En tout il y a le Kilobot basique qui va nous servir de base de comparaison pour tout ces tests, 3 Kilobots dont les pattes ont été coupées et remplacées par des pattes de différentes formes et matériaux et 4 Kilobots auxquels on a ajouté des exosquelettes tous différents les uns des autres.

Ci-dessous voici une vue isométrique de chaque prototype, avec ses propres spécificités.

	<p><u>Double brosse à dent</u></p> <p>Kilobot auquel on a enlevé ses 3 pattes d'origine, pour les remplacer par 2 têtes de brosse à dent parallèles.</p>
	<p><u>Triple brosse à dent</u></p> <p>Kilobot auquel on a enlevé ses 3 pattes d'origine, pour les remplacer par 3 têtes de brosse à dent, placées aux mêmes endroits que les pattes d'origine</p>
	<p><u>Bloc circulaire</u></p> <p>Exosquelette avec un arc de cercle en guise de pattes arrières, sa patte avant est en forme de pointe pour créer un contact ponctuel au niveau de la surface, tout en ayant une grande robustesse.</p>
	<p><u>Bande arrière courte</u></p> <p>Exosquelette avec une bande arrière de petite taille en guise de pattes arrières, sa patte avant est un parallélépipède rectangle, elle ne crée pas de contact ponctuel avec la surface mais est néanmoins robuste.</p>
	<p><u>Bande arrière large</u></p> <p>Exosquelette avec une bande arrière de grande taille en guise de pattes arrières, sa patte avant est un parallélépipède rectangle, elle ne crée pas de contact ponctuel avec la surface mais est néanmoins robuste.</p>
	<p><u>Grenouille</u></p> <p>Exosquelette avec des pattes à la même dispositions que les pattes du Kilobot, à la différence que celles de l'exosquelette ne sont pas des contacts ponctuels.</p>

FIGURE 2 – Descriptif des prototypes d'exosquelettes déjà conçus par les membres de l'ISIR

L'objectif va être, via différents protocoles expérimentaux, d'observer les qualités et défauts de ces 7 prototypes avec comme base le Kilobot basique. Parmi les qualités pertinentes que nous avons déterminés, on peut citer la stabilité du Kilobot, sa vitesse de déplacement en translation, sa vitesse de rotation et surtout sa trajectoire. Pour cette dernière c'est quelque chose de très important car avant de travailler sur un correcteur de trajectoire, il faut s'assurer que le Kilobot est capable de suivre au mieux une ligne droite, lorsque l'on met la même puissance de vibration dans chaque moteur. De ce fait dès le début, on limite considérablement le travail que doit accomplir le correcteur de trajectoire, en s'attellant à créer le prototype avec la meilleure trajectoire naturelle.

Ainsi on va faire une première batterie de tests purement visuelle, avec une note / 20 attribuée à chaque Kilobot pour chacune des caractéristiques citées dans le paragraphe précédent. Par purement visuelle nous entendons que nous allons évaluer

de manière assez rapide quels sont les prototypes que nous décidons de ne pas garder, car peu intéressants concernant la création de nos propres prototypes.

De ce fait on va les faire aller tout droit et observer leur vitesse de déplacement et s'ils vont bien tout droit, on va leur faire faire une rotation simple et voir s'ils tournent rapidement et sur eux même (s'ils tournent autour d'un axe qui n'est pas l'axe au centre du robot ça n'est pas très bon), on va aussi évaluer leur stabilité en les confrontants à de petits chocs (un autre Kilobot qui entre en collision à droite, à gauche, à l'avant et à l'arrière du prototype que l'on teste).

Une fois que nous avons évalué chacun des prototypes, nous avons classés toutes ces notes dans le tableau ci-dessous et tiré certaines conclusions de nos observations. Nous avons également fait quelques suppositions sur la cause de certains défauts des prototypes. A savoir que les notes attribuées sont les évaluations des robots dans leurs conditions optimales (un prototype va bien mieux marcher à une certaine puissance plutôt qu'à une autre par exemple).

Remarque : Les puissances des vibreurs vont de 0 à 255, avec 0 une puissance nulle et 255 la puissance maximale. On notera que dans les deuxième, troisième et quatrième colonnes en partant de la gauche les P signifient puissance, avec P qui est donc compris entre 0 et 255.

Test Macro	Vitesse de déplacement (entre 80, 100)	Trajectoire	Vitesse de rotation	Equilibre statique
Basique	15/20 P = 80	P > 80 : Biaisée P < 80 : Bonne	16/20 P = 70 Lent et précis	Axial : 8/10 Latéral : 8/10
Brosse à dent - Double	20/20 P = peu importe	Bonne pour n'importe quelle puissance	20/20 P = 100 Rapide et précis	Axial : 4.5/10 Latéral : 9/10
Brosse à dent - Triple	7/20 P = 100	Mauvaise	4/20 Mauvaise	Axial : 4/10 Latéral : 4/10
Caoutchouc	15/20 P = peu importe	Bonne pour n'importe quelle puissance	15/20 Rot G bancale, Rot D bonne	Axial : 6/10 Latéral : 5/10
Bloc circulaire	9/20 P = 100	Mauvaise	18/20 P = 100 Lent et précis	Axial : 10/10 Latéral : 9/10
Bande arrière - courte	11/20 P = 100	Fiable à faible puissance	11/20 P = 90 Lent et tourne pas sur lui même	Axial : 8/10 Latéral : 6/10
Bande arrière large	10/20 P = 100	Fiable à faible puissance	14/20 P = 90 Moyenne et tourne pas sur lui même	Axial : 8/10 Latéral : 8/10
Position Grenouille	11/20 P = 100	Fiable à faible puissance	12/20 P = 90 Lent et tourne pas sur lui même	Axial : 7/10 Latéral : 5.5/10

FIGURE 3 – Tableau des tests macroscopiques sur les prototypes première génération

### 4.3.2 Premiers bilans

## Bilan individuel

Basique : Le robot est réactif et précis mais simplement sur certaines plages de puissance (entre 60 et 80). Pour des travaux de précisions sans contrainte de temps, c'est adapté, sinon : à améliorer.

Double brosse à dents, modification : À basse puissance on remarque que le Kilobot tremble un peu, ce qui n'est pas le cas à haute vitesse. On émet l'hypothèse que lors de la vibration les fils des brosses à dents rentrent en phase et imposent leur propre vibration, désynchronisée de la vibration des moteurs, ce qui rend ce modèle peu précis à basse puissance ais très efficace à haute puissance.

Caoutchouc, modification : Pour ce qui est de la rotation on remarque qu'elle est parfois imprécise. On émet l'hypothèse que cela est sûrement dû aux coefficients de frottement du caoutchouc : les pieds accrochent à la surface et freinent la rotation. Mis à part cela ce prototype n'a aucun problème en translation. Cependant le gros soucis est la stabilité du robot malgré un centre de gravité bas (bas par rapport aux autres robots qui sont plus stables).

Bloc circulaire : Au niveau des pattes il y a trop de surface d'application et trop de poids, le petit robot a du mal à déplacer une si grosse structure. En terme de rotation il est bon, mais peut être un peu trop lent.

## Bilan comparatif

- La surface des pattes arrières doit être comprise entre la surface des pattes arrières de l'exosquelette bande courte et celle des pattes de l'exosquelette grenouille. De plus l'exosquelette ne doit pas dépasser un certain volume (trop de ralentissements sinon).
- L'exosquelette bande arrière large tourne mieux que l'exosquelette bande arrière courte car le centre d'inertie est plus proche du centre de gravité du robot (ceci est dû à la bande plus large et plus proche du centre).
- L'équilibre latéral est meilleur dans le cas de pattes en forme de bandes continues ; la stabilité est meilleure là où le centre de gravité est au centre du robot : donc les exosquelettes bloc circulaire et bande large sont bons dans ce domaine. En conclusion il vaut mieux privilégier les systèmes à pattes éloignées.

## Bilan autre

- Il est préférable d’avoir deux pattes arrières minimum, on déduit cela du fait qu’à chaque fois que nous avons une seule grande patte arrière, le Kilobot deviens moins fiable en terme de direction.
- Les capteurs infra-rouge bénéficient de moins de visibilité pour l’exosquelette bande arrière large que pour l’exosquelette bande arrière courte, c’est un problème pour le transfert de données, à prendre en compte pour la conception des prototypes deuxième génération.

### 4.3.3 Finalisation de l’analyse

De toutes les conclusions que nous avons tirés dans les trois bilans précédents, nous avons déjà une idée de certains types de prototypes à faire, mais pour préciser et approfondir notre idée du prototype le plus optimisé possible, il faut également faire une étude concernant les dimensions de l’exosquelette et plus généralement concernant toutes les cotations optimales qu’on ferait lors de la modélisation 3D des prototypes.

Pour cela on a choisi de garder seulement les exosquelettes, car pour l’instant il est impossible de faire de l’impression 3D directement sur un Kilobot. Il faut donc créer un module externe à celui-ci, dans lequel le robot viendrait s’emboîter directement.

Nous avons donc classé dans le tableau ci-dessous les dimensions de tous les exosquelettes déjà construits, ces dimensions comprennent : l’écart entre les deux pattes arrières, l’écart diagonal entre les pattes arrières et la patte avant, le diamètre du robot et la longueur des pattes, et le diamètre des trous pour accueillir les pattes du Kilobot lorsqu’il va s’emboîter dans l’exosquelette.

On cherchera en sommes ici à comparer les dimensions des exosquelettes première génération avec celle du Kilobot Basique pour définir les écarts qui sont dans une plage acceptable et ainsi les dimensions des exosquelettes qui sont adaptées.

Mesures	Écart pattes arrières (cm)	Écart pattes diagonales (cm)	Diamètre trou Pattes (mm)	Longueur patte (cm)	Diamètre patte imprimée (mm)
Basique	2.55	2.80	Pas de trou	1.9 - 1.95	Pas de patte
Bloc circulaire	2.65	2.88	2.0	1.94 - 1.95	Pas de patte
Grenouille	2.60	2.75	2.5	1.96	0.8
Bande Large	2.70	2.70	2.5	1.95	0.89
Bande Courte	2.60	2.75	2.5	1.95	0.82

FIGURE 4 – Tableau affichant les cotes principales des exosquelettes première génération

## Bilan des mesures pour la modélisation

---

- Il faut un écart minimum entre les 3 trous à faire dans les exosquelettes, pour laisser passer les pattes du Kilobot, il faut que ce soit très précis car si les trous ne sont pas à bonne distance les uns des autres, on devra forcer pour faire rentrer le Kilobot et les pattes de celui-ci risqueraient de casser. C'est une contrainte immuable.
- Concernant les trous qui vont accueillir les pattes du Kilobot, il faut que leur diamètre soit assez large pour être sûr que la patte puisse rentrer, mais pas trop large afin de limiter les pertes d'énergie. En effet plus le robot sera immobile dans son socle, plus les forces de vibration seront bien transmises aux pattes de l'exosquelette, et ainsi celui-ci ne verra pas ses capacités trop diminuées.
- L'écart diagonal entre les pattes doit être de  $2.8 \pm 0.05$  cm. Il faut veiller à respecter le  $\pm 0.05$  cm, car sur les deux socles : bloc circulaire et bande large, on a un écart de  $2.8 \pm 0.07$  cm, et on remarque une forte déformation au niveau des pattes du Kilobot lorsqu'on le fait rentrer dans ces deux supports. On estime alors à  $\pm 0.05$  cm l'écart acceptable.
- L'écart entre les deux pattes arrières doit être de  $2.55 \pm 0.04$  cm, et ceci pour des raisons équivalentes à celles citées pour l'écart diagonal.

De tout ceci on remarque que nous sommes assez contraints concernant la stabilité du robot. En effet, si on veut s'assurer une stabilité parfaite, on fait en sorte que les pattes de l'exosquelette soient le plus éloignées les unes des autres possible, or il n'est pas concevable de les éloigner plus qu'elles ne le sont déjà sur le Kilobot de base sans perdre niveau performance. Il faut donc trouver un compromis entre performance et stabilité, compromis que nous retrouvons dans les prototypes que nous avons créés, dont nous allons parler ci-dessous, dans la partie consacrée exclusivement à ceux-ci.

## 5 Création de nos propres exosquelettes et étude bien plus poussée de ceux-ci

En fonction de toutes les contraintes que nous avons énoncées dans la partie précédente, nous avons été en mesure de créer 3 prototypes d'exosquelettes. Un quatrième module très prometteur a quand à lui été créé par un post-doctorant du nom de Matan Yah Ben durant notre stage, nous avons donc décidé de l'ajouter à notre panel d'exosquelettes afin de le tester également. Donc, une fois les premiers schémas dessinés, il a fallu les modéliser sur le logiciel de modélisation 3D Catia. Tous nos prototypes devaient être modélisés virtuellement car il fallait les imprimer grâce à une imprimante 3D. En effet si l'une des grandes qualités du Kilobot est qu'il est facilement reproductible et peu cher, toute amélioration qui lui est apportée ne doit pas changer cette ligne directrice [14]. L'impression 3D, étant une méthode qui



peut s'utiliser dans l'industrie et la production en série, et utilisant des matériaux peu coûteux, elle semble être une solution très viable dans notre cas.

Il faut également savoir que tous les exosquelettes ont été imprimés à l'ISIR ou au Fablab de Sorbonne Université, sauf l'exosquelette "Spider" qui a été imprimé à l'ESPCI (par Matan Yah Ben qui est post-doctorant sur le projet).

Ainsi la modélisation à l'échelle de ces 3 exosquelettes est la base de cette grande partie de création de modules, sans quoi nous n'aurions pas pu mener les études détaillées les concernant, car on le rappelle le but est de tester les Kilobots dans les conditions physiques réelles, sans tomber dans les approximations de celles-ci.

## 5.1 Présentation des différents exosquelettes

Voici donc une description complète des 4 exosquelettes seconde génération :

### 5.1.1 L'exosquelette "Brosse à dent"

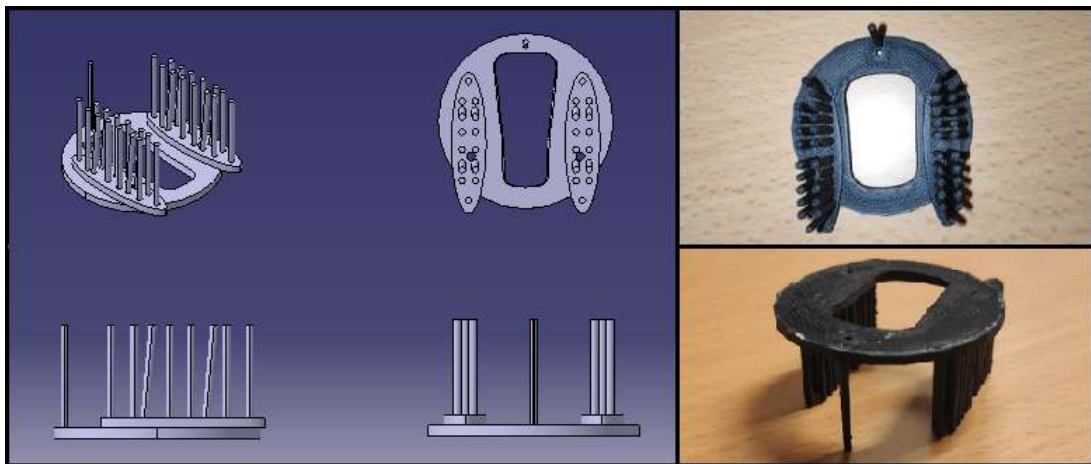


FIGURE 5 – Vues isométriques de l'exosquelette "Brosse à dent" modélisé sous Catia (à gauche), imprimé (à droite)

Ce module est un exosquelette brosse à dent et non une modification brosse à dent comme dans la première partie, il est important de les distinguer. Pour la modification brosse à dent nous avons bien enlevé les pattes du Kilobot et collé des têtes de brosse à dent directement sur celui-ci. Pour l'exosquelette, la définition de ce mot l'explique assez bien, mais c'est une structure externe au Kilobot non-estropié, qu'on clipse sur celui-ci, dont les pattes ont la même disposition "ovale" que pour les fils d'une brosse à dent.

La pertinence du fait d'avoir créé un module de ce type, viens du fait des nombreuses qualités que la modification brosse à dent avait sur le Kilobot : sa vitesse était grandement augmentée, sa précision de trajectoire à haute vitesse était impeccable et sa vitesse de rotation plus que convenable.

Ceci dit, on le rappelle, sa stabilité axiale (le fait qu'il basculait très facilement vers l'avant) n'était pas optimale et surtout nous avons émis l'hypothèse que la vibration des fils de nylon de la brosse à dent rendait son comportement instable



à basse puissance. Donc pour vérifier cette théorie et pallier au soucis de stabilité axiale, il nous a semblé judicieux de déterminer si cela était bien dû à la proximité que chacun des fils de nylon de la brosse à dent avait avec les autres. Pour cela nous avons créé ce prototype qui, dû au fait que les fils de nylon soient remplacés par des tiges de plastique beaucoup plus rigides et espacées, ne devrait en théorie pas entrer dans cette phase d'instabilité à basse puissance. Si notre théorie s'avère être correcte, nous pourrions ainsi établir que le défaut d'instabilité est simplement dû à la disposition des pattes et/ou au matériau qui compose celles-ci.

C'est un raisonnement par élimination : on étudie le comportement du robot en changeant à chaque fois une caractéristique, ce qui nous indique qu'est ce qui est à l'origine d'un dysfonctionnement potentiel.

Par ailleurs un autre problème à régler était la stabilité (au sens physique du terme) de notre Kilobot double brosse à dent : il avait tendance à basculer facilement vers l'avant et c'est un de ses principal défauts. Pour palier à cela nous avons installé une patte supplémentaire à l'avant de notre exosquelette, en effet nous avons déterminé précédemment que pour une stabilité optimale il fallait 3 pattes éloignées le plus possible les unes des autres.

### 5.1.2 L'exosquelette "Triangle penché"

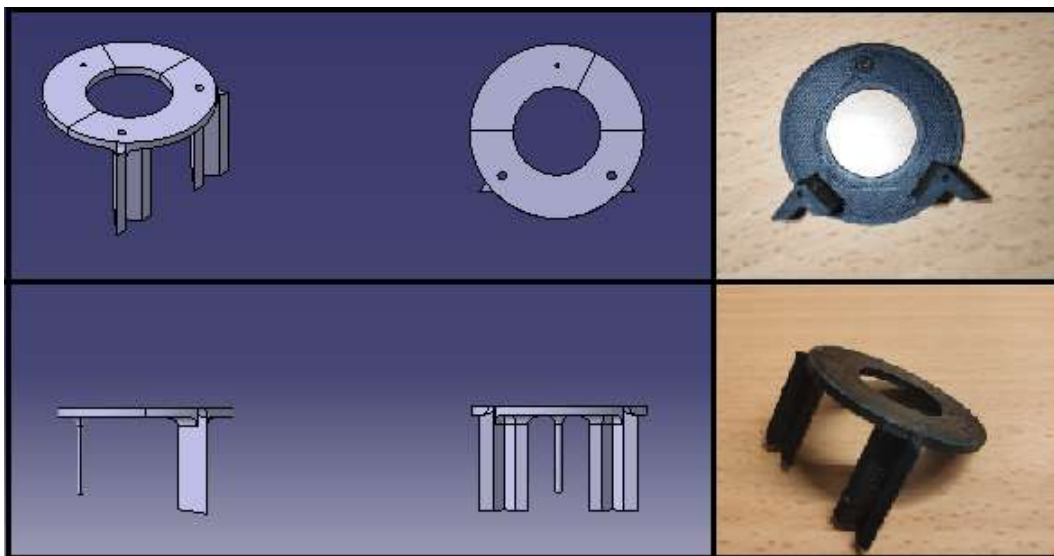


FIGURE 6 – Vues isométriques de l'exosquelette "Triangle penché" modélisé sous Catia (à gauche), imprimé (à droite)

La pertinence sur la création de ce prototype réside dans le fait que nous avons observé que les modules dont les pattes étaient légèrement penchées vers l'avant allaient plus vite, on peut citer le "Double brosse à dent" dont les fils de nylons étaient penchés vers l'avant, et qui possédait une vitesse de déplacement exceptionnelle.

Il possède donc deux pattes arrières de sections triangulaires, qui sont tronquées de sorte que lorsque le Kilobot équipé de l'exosquelette est posé à plat sur une

surface, il soit penché vers l'avant. Par ailleurs le choix d'avoir fait des pattes de section triangulaire viens du fait que nous avons émis l'hypothèse que peut être si on reprennait le concept qui permet au Kilobot d'avancer (c'est à dire avoir une surface d'application asymétrique), on pourrait compenser le freinage que le remplacement de pattes ponctuelles par des pattes en contact surfacique avec le terrain de déplacement imposait. Cela reste encore une hypothèse que nous avons jugé bon de tester.

Ainsi ce prototype mérite qu'on y porte un intérêt certain, et au moment de sa création nous sommes très optimistes quand à son bon fonctionnement.

### 5.1.3 L'exosquelette "Courbé"

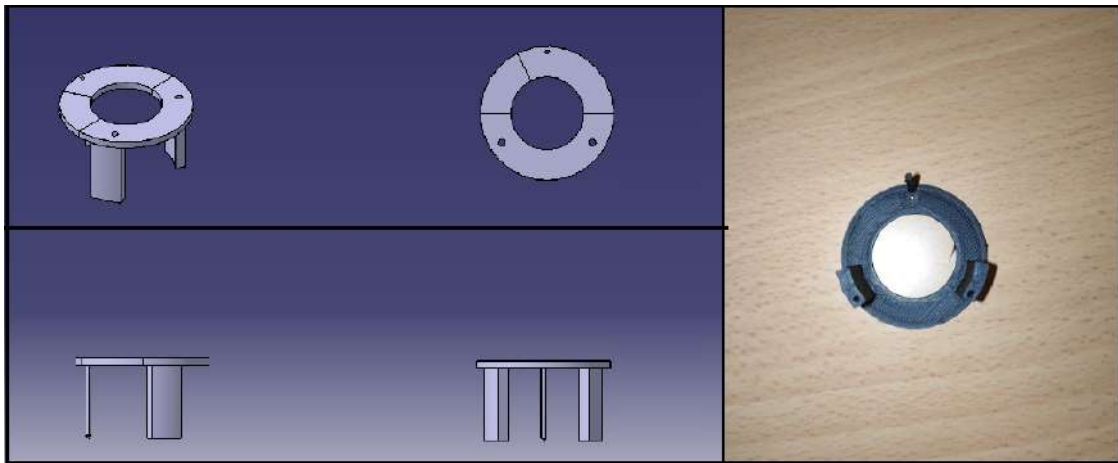


FIGURE 7 – Vues isométriques de l'exosquelette "Courbé" modélisé sous Catia (à gauche), imprimé (à droite)

Ce prototype à été créé pour calquer les propriétés de l'exosquelette "Bloc circulaire" qui sont la stabilité et la précision lors d'une rotation, mais sans les défauts de vitesse de déplacement et de rotation. Nous avons déduis que tous ces problèmes de vitesse étaient liés au caractère imposant de l'exosquelette, et surtout au fait qu'il n'avais pas deux pattes arrières. Donc nous avons simplement fait d'une pierre deux coups en créant un espace entre les deux extrémités de l'arc de cercle, ce qui crée donc deux pieds en forme d'arc de cercle du même rayon que pour "Bloc circulaire".

On espère avec cela au moins réduire son poids suffisamment pour qu'il puisse avancer et tourner à vitesse convenable.

#### 5.1.4 L'exosquelette "Spider"



FIGURE 8 – Vue isométrique de l'exosquelette "Spider" imprimé

Cet exosquelette est, comme nous l'avons dit, le seul qui nous a été mis à disposition pour l'étude des exosquelettes deuxième génération et le seul qui n'a pas été conçu et imprimé à l'ISIR mais à l'ESPCI. De ce fait nous n'avons aucune image concernant la modélisation 3D de celui-ci.

Par ailleurs il est intéressant car nous allons pouvoir vérifier notre théorie concernant la perte d'énergie due au fait que les pieds du Kilobot ne soient pas maintenus immobiles et entourés par l'exosquelette. Il est difficile de le décrire physiquement mais on remarque déjà qu'il dispose de 4 pattes et non 3, par ailleurs celles-ci sont très éloignées les unes des autres comparés aux autres modules, ce qui nous permet d'annoncer une bonne stabilité, avant même de l'avoir testé.

## 5.2 Protocoles de tests des exosquelettes

La conclusion de cette première grande partie concerne le test des exosquelettes que nous avons créés, et pour faire ces tests efficacement, de manière répétée et sans se tromper dans les manipulations à effectuer, nous avons rédigés deux protocoles. Le premier va servir à déterminer la vitesse de rotation de chacun des modèles, et le second servira à déterminer la précision et la vitesse de déplacement des robots.

En ce qui concerne le matériel nécessaire, nous avons eu besoin d'une grande arène faite exclusivement pour le test des Kilobots, car tapissée d'une surface réfléchissante dont l'utilité a déjà été expliquée précédemment (figure n°4, un peu plus bas). De plus cette arène peut servir pour positionner la caméra ou le téléphone qui servira à filmer nos tests. On avait également besoin d'un logiciel de pointage vidéo, nous avons utilisé LatisPro mais il en existe bien plus, compatibles avec toutes les plateformes et systèmes d'exploitations. Nous avons besoin des Kilobots chargés ainsi que des exosquelettes que nous voulions tester, d'un laser et d'un mètre.

### 5.2.1 Protocole de calcul de vitesse de rotation

- Pour calculer la vitesse de rotation d'un Kilobot il faut tout d'abord compiler le programme dans celui-ci qui va envoyer l'instruction de vibrer à la puissance 80 (toujours de 0 à 255 la puissance on rappelle) au moteur de notre choix (on choisit le moteur droit).
- L'autre moteur ne fait rien.
- On filme d'une vue de dessus le robot en plaçant une caméra ou un téléphone au dessus de celui-ci, à hauteur raisonnable (de l'ordre du mètre), bien parallèle à la surface sur laquelle est posé le Kilobot et immobile surtout.
- Après avoir implémenté le programme sur le Kilobot, on active celui-ci, on laisse le robot faire 4 à 5 tours sur lui même et on éteint l'enregistrement de la vidéo.
- Ensuite tout se passera sur ordinateur. On utilise notre logiciel de pointage vidéo et via ce logiciel on trace une ligne virtuelle sur la vidéo, cela nous servira de repère.
- On localise un point fixe sur le Kilobot, et on compte le nombre de fois ou ce point fixe traverse la ligne virtuelle qu'on a tracé juste avant. On relève par la suite le temps qui sépare la première fois ou le point traverse la ligne et la dernière fois ou le point la traverse.
- Ainsi on a un certain nombre de tours par un certain nombre de secondes, ce qui après conversion nous donne une vitesse de rotation du Kilobot en tour/minute.

### 5.2.2 Protocole de calcul de trajectoire et vitesse de déplacement

Pour ce protocole il faut savoir que la manipulation est la même pour le calcul de la trajectoire et le calcul de la vitesse de déplacement, en fait c'est sur le logiciel de pointage qu'on va déterminer ces deux données différentes. C'est là la grande utilité de ce genre de logiciels, car en une vidéo on peut déterminer un certain nombre de données différentes.

- Il faut tout d'abord compiler le programme sur le microcontrôleur qui va envoyer l'instruction de vibrer à la même puissance les deux moteurs [2]. Pour diversifier nos mesures, on va faire vibrer les moteurs à 3 puissances différentes : 70, 120 et 250 ; on fait cela car pour certains prototypes, certaines puissances rendent le comportement du prototype en question un peu aléatoire, ce qui n'est pas très utile dans le cadre de nos recherches.
- On place un laser qui va tracer sur l'arène de test une ligne qui modélise le chemin que le Kilobot devrait prendre si il avait une trajectoire parfaite.
- On met le zéro du mètre à l'endroit où on veut faire partir le Kilobot lors du test, et on le déroule de 50 cm en le mettant bien parallèle au faisceau du laser.
- On filme d'une vue de dessus le robot en plaçant une caméra ou un téléphone au dessus de celui-ci, à hauteur raisonnable (de l'ordre du mètre), bien parallèle à la surface sur laquelle est posé le Kilobot et immobile surtout.
- Après avoir implémenté le programme sur le Kilobot, on place celui-ci bien

droit sur faisceau laser au niveau du zéro du mètre, et on l'active. Une fois que le robot arrive au bout des 50 cm, on peut éteindre l'enregistrement de la vidéo.

- On pointe ensuite la position du Kilobot sur la vidéo qu'on a pris via notre logiciel de pointage. Ceci va nous donner une position en X et Y en fonction du temps, ce qui va nous permettre de tracer la trajectoire des robots sur un graphique. On pourra également déterminer la vitesse et l'accélération moyenne des Kilobots lors de leur trajet, en récupérant la norme des positions à chaque point qu'on a effectué et en faisant leur moyenne.
- On réitère l'opération 2 fois en changeant à chaque fois la puissance allouée aux vibreurs des Kilobots.

Ci-dessous voici le schéma explicatif sur la disposition de tous les éléments :

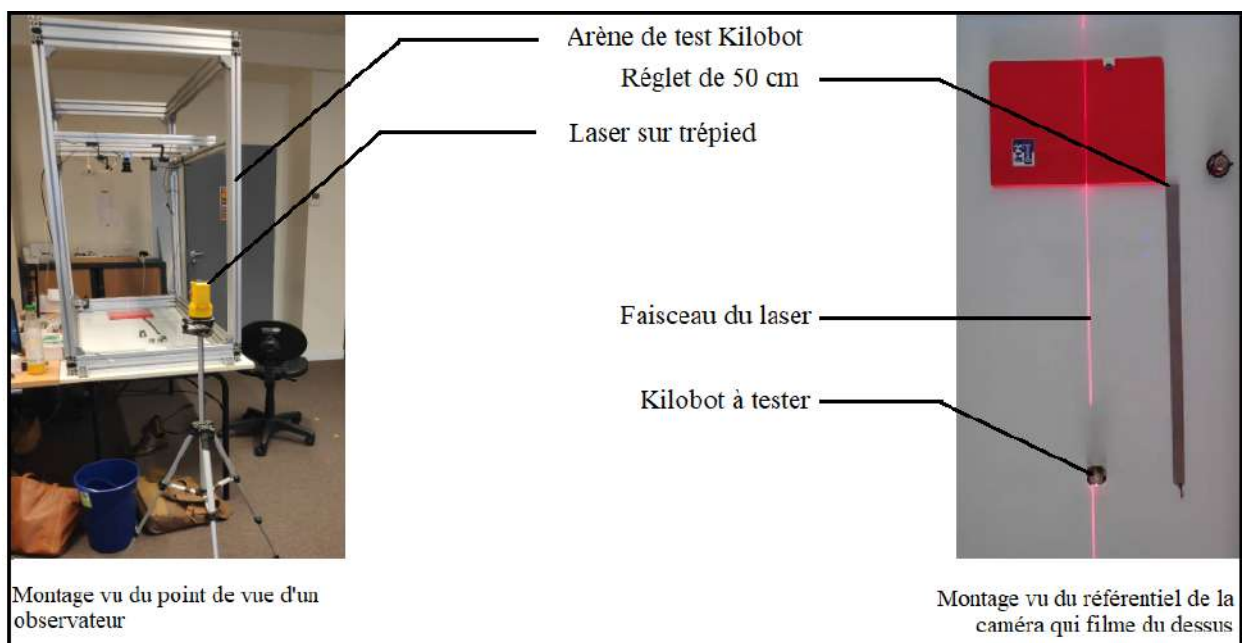


FIGURE 9 – Schéma explicatif sur la disposition des éléments de notre montage

### 5.3 Résultats de nos tests

Une fois que nous avons appliqués les deux protocoles que nous avons établis, à chacun des exosquelettes qu'on a choisi d'étudier et au Kilobot seul (il nous servira de base), on peut faire une interprétation des graphes et données qu'on a recueillis grâce à ceux-ci.

Par ailleurs, nous avons décidé d'appliquer les protocoles précédents sur le Kilobot "Double brosse à dent, modification" en plus des exosquelettes que nous avons créés, car bien qu'il n'ait pas été créé par nos soins, on estime qu'il remplit si bien sa fonction que nous nous devons de faire la même étude poussée sur lui que nous faisons sur nos propres exosquelettes. C'est le seul Kilobot modifié que nous avons décidé de garder pour l'étude poussée, ce qui appuie encore une fois le fait qu'il possède des qualités que nous souhaitons à tout prix évaluer.

### 5.3.1 Graphes de trajectoires

Ci-dessous voici les trajectoires que chacun des robot à pris durant les différents tests qu'on a faits ainsi que l'interprétation associée à chacun de ceux-ci.

Remarque : pour tous les graphiques l'axe des abscisses est la représentation du faisceau laser.

Kilobot classique :

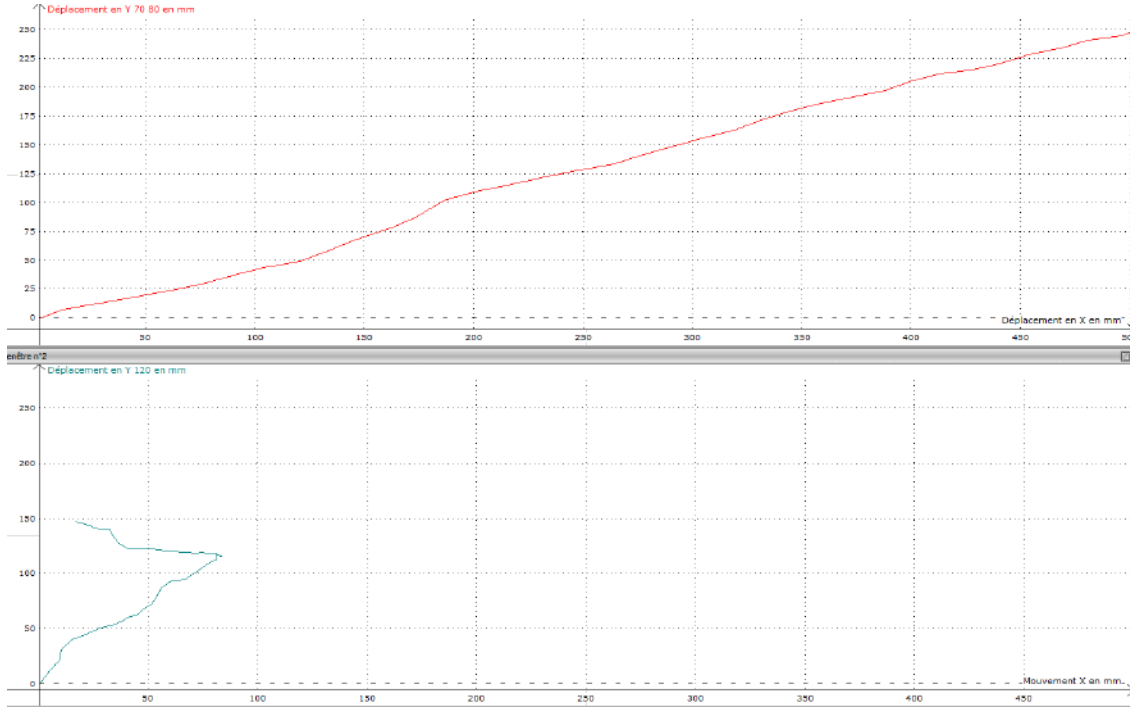


FIGURE 10 – Graphique montrant les trajectoire du Kilobot basique, avec  $P_{moteurs} = 80$  (graphe du haut) et  $P_{moteurs} = 120$  (graphe du bas)

Pour le Kilobot seul, nous avons choisi d'afficher les deux puissances de vibrations  $P_{moteurs} = 80$  et  $P_{moteurs} = 120$ , car déjà à partir de  $P_{moteurs} = 120$  on remarque que le Kilobot va dans tous les sens et cela n'a aucun intérêt de représenter sa trajectoire sur un graphique. Pour le Kilobot basique nous avons représenté ce graphique qui montre la puissance "limite", pour les autres exosquelettes dont le comportement sera similaire (car il y en a) on n'affichera pas les graphes concernés, et on ne calculera non plus pas la vitesse du Kilobot à la puissance concernée.

On voit très clairement que plus  $P_{moteurs}$  augmente, plus le Kilobot voit sa trajectoire déviée. On va donc privilégier une utilisation à basse puissance pour le Kilobot seul.

→ Sa trajectoire sur un demi-mètre est plutôt mauvaise peu importe la valeur de  $P_{moteurs}$ , ceci dit il se déplace quand même vers l'avant, nous prenons donc cela en compte.

Ainsi l'écart entre le point d'arrivée pour une trajectoire parfaite et le point d'arrivée pour nos trajectoires réelles, sur une distance de 0.5 m, est :  $e_{80} = 0.250 \text{ m}$ .

Pour ce qui est du relevé de vitesse de déplacement, on trouve :  $V_{80} = 0.008 \text{ m.s}^{-1}$ .

Pour finir, pour la vitesse de rotation on a :  $V_{rot80} = 7.5 \text{ tr/min}$

Brosse à dent double, modification :

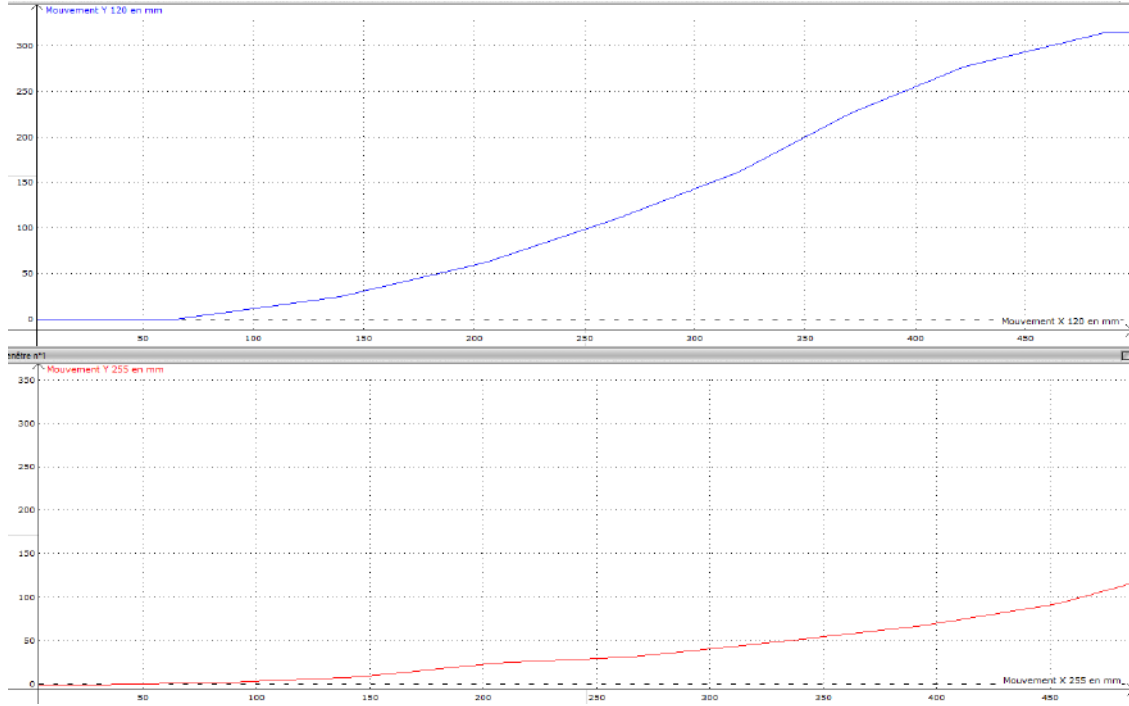


FIGURE 11 – Graphique montrant les trajectoire du Kilobot "Brosse à dent double, modification", avec  $P_{moteurs} = 120$  (graphe du haut) et  $P_{moteurs} = 255$  (graphe du bas)

Pour ce Kilobot, nous avons choisi d'afficher les deux puissances de vibrations  $P_{moteurs} = 120$  et  $P_{moteurs} = 255$ , car en dessous de la  $P_{moteurs} = 120$  le Kilobot allait dans tous les sens et, on le repète, cela n'avait aucun intérêt de représenter sa trajectoire sur un graphique. On voit très clairement que plus  $P_{moteurs}$  augmente, plus le Kilobot à une trajectoire rectiligne ce qui est exactement ce que nous lui demandons.

→ Globalement sa trajectoire sur un demi-mètre est correcte à  $P_{moteurs} = 120$ ,  $P_{moteurs} = 255$ , et plus généralement sur l'intervalle que ces deux puissances forment.

Ainsi l'écart entre le point d'arrivée pour une trajectoire parfaite et le point d'arrivée pour nos trajectoires réelles, sur une distance de 0.5 m, sont :  $e_{120} = 0.300 \text{ m}$  et  $e_{255} = 0.110 \text{ m}$

Pour ce qui est des relevés de vitesse de déplacement, on trouve :  $V_{120} = 0.098 \text{ m.s}^{-1}$  et  $V_{255} = 0.3 \text{ m.s}^{-1}$ .

Pour finir, pour la vitesse de rotation on a :  $V_{rot80} = 14 \text{ tr/min}$



Courbé, exosquelette :

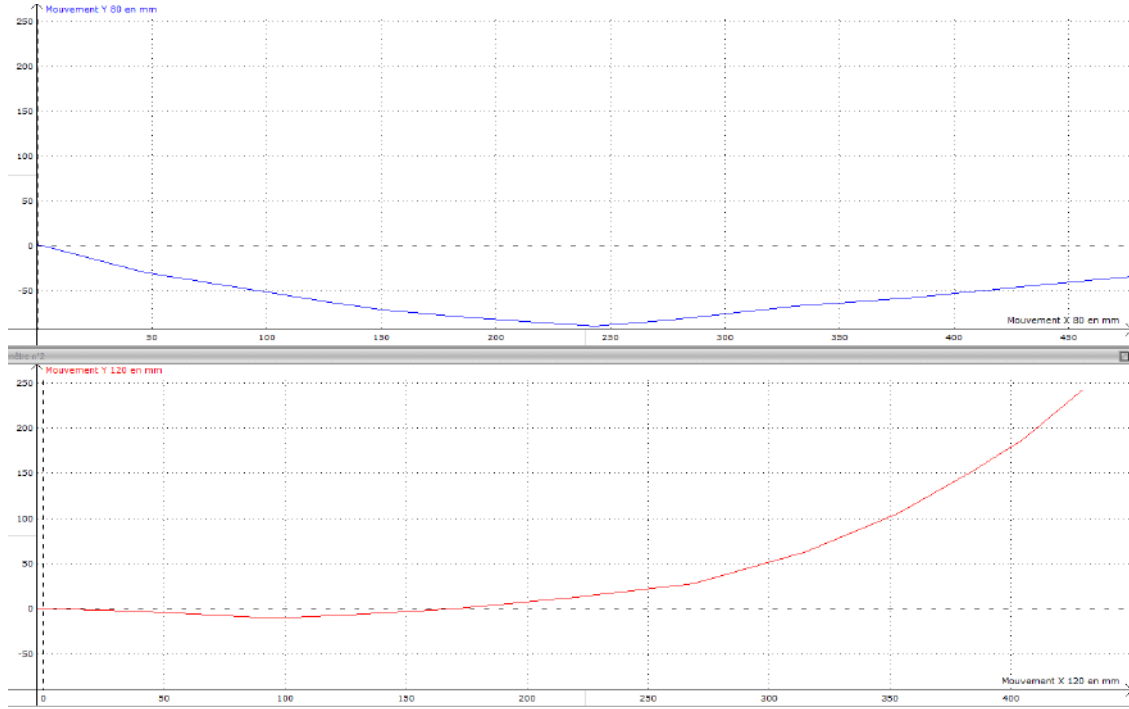


FIGURE 12 – Graphique montrant les trajectoire du Kilobot équipé de l'exosquelette "Courbé", avec  $P_{moteurs} = 80$  (graphe du haut) et  $P_{moteurs} = 120$  (graphe du bas)

Pour cet exosquelette, nous avons choisi d'afficher les deux puissances de vibrations  $P_{moteurs} = 80$  et  $P_{moteurs} = 120$ , car au delà de la  $P_{moteurs} = 120$  le Kilobot allait encore une fois dans tous les sens. On voit très clairement que plus  $P_{moteurs}$  augmente, plus le Kilobot voit sa trajectoire déviée. On va donc privilégier une utilisation à basse puissance pour ce modèle.

→ Sa trajectoire sur un demi-mètre est très correcte à  $P_{moteurs} = 80$  mais au delà elle est mauvaise.

Ainsi l'écart entre le point d'arrivée pour une trajectoire parfaite et le point d'arrivée pour nos trajectoires réelles, sur une distance de 0.5 m, sont :  $e_{80} = 0.030 \text{ m}$  et  $e_{120} = 0.270 \text{ m}$

Pour ce qui est des relevés de vitesse de déplacement, on trouve :  $V_{80} = 0.0066 \text{ m.s}^{-1}$  et  $V_{120} = 0.064 \text{ m.s}^{-1}$ .

Pour finir, pour la vitesse de rotation on a :  $V_{rot80} = 3.7 \text{ tr/min}$



### Brosse à dent double, exosquelette :

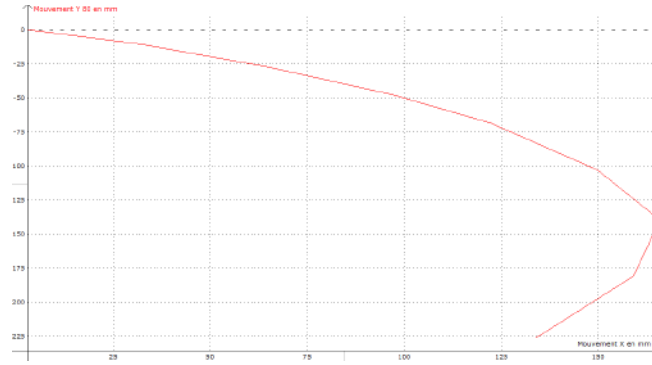


FIGURE 13 – Graphique montrant la trajectoire du Kilobot équipé de l'exosquelette "Brosse à dent", avec  $P_{moteurs} = 80$

Concernant ce prototype, nous avons été très étonnés lorsque nous nous sommes aperçus que sa trajectoire n'était bonne à aucune puissance. Nous avons néanmoins décidé d'afficher un des graphiques modélisant sa trajectoire, mais il faut savoir qu'à n'importe quelle puissance le Kilobot n'a jamais atteint le bout du trajet de 50 cm.

Même concernant sa rotation, il essayait de tourner du mieux qu'il pouvait mais il ne le faisait pas bien, très lentement, et surtout pas sur place. Nous mettions beaucoup d'espoir dans cet exosquelette, mais malheureusement il n'a pas les qualités que nous lui prédisions.

Ceci dit nous avons quand même pu déterminer quelques données à son sujet, notamment sa vitesse de rotation  $V_{rot80} = 1.3 \text{ tr/min}$ , et sa vitesse de déplacement  $V_{80} = 0.019 \text{ m.s}^{-1}$

### Triangle penché, exosquelette :

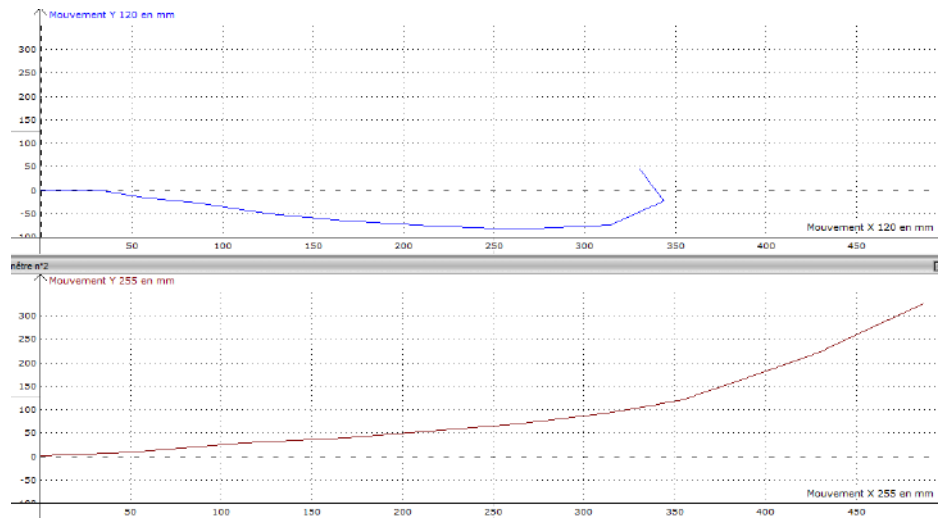


FIGURE 14 – Graphique montrant les trajectoires du Kilobot équipé de l'exosquelette "Triangle penché", avec  $P_{moteurs} = 120$  (graphe du haut) et  $P_{moteurs} = 255$  (graphe du bas)

Pour cet exosquelette, nous avons choisi d'afficher les deux puissances de vibrations  $P_{moteurs} = 120$  et  $P_{moteurs} = 255$ , car à  $P_{moteurs} = 120$  et en dessous le Kilobot ne va plus dans la direction vers laquelle on veut qu'il aille. Cependant même sachant cela, pour  $P_{moteurs} = 120$  il dévie trop de sa position mais très lentement (il n'y a pas de gros virage imprévisible, c'est plus un défaut de calibration) donc on peut quand même calculer sa vitesse de déplacement. Cependant on remarque que plus on augmente  $P_{moteurs}$  plus la trajectoire s'améliore, même si celle-ci demeure médiocre.

→ Sa trajectoire sur un demi-mètre est donc plutôt mauvaise peu importe la valeur de  $P_{moteurs}$  mais elle tend à s'améliorer en fonction proportionnellement à celle-ci.

Ainsi l'écart entre le point d'arrivée pour une trajectoire parfaite et le point d'arrivée pour nos trajectoires réelles, sur une distance de 0.5 m, est :  $e_{255} = 0.325 \text{ m}$ .

Pour ce qui est des relevés de vitesse de déplacement, on trouve :  $V_{120} = 0.027 \text{ m.s}^{-1}$  et  $V_{255} = 0.1 \text{ m.s}^{-1}$ .

Spider, exosquelette :

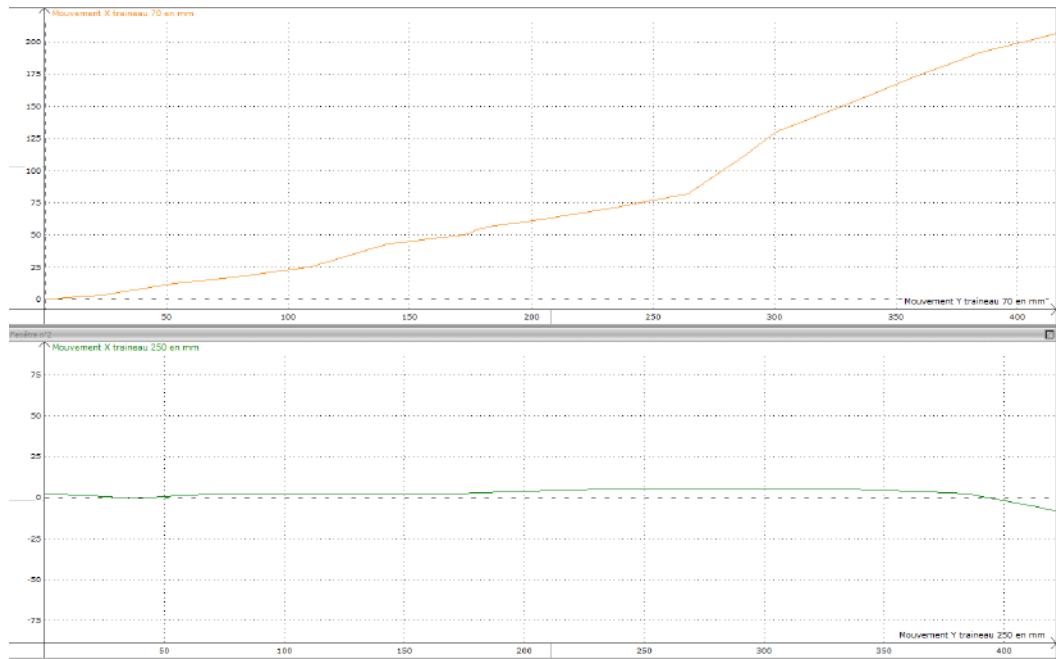


FIGURE 15 – Graphique montrant les trajectoire du Kilobot équipé de l'exosquelette "Spider", avec  $P_{moteurs} = 80$  (graphe du haut) et  $P_{moteurs} = 255$  (graphe du bas)

Pour cet exosquelette, nous avons choisi d'afficher les deux puissances de vibrations  $P_{moteurs} = 80$  et  $P_{moteurs} = 255$ , car sur toutes les plages de puissance il n'a pas de comportement ou il fait n'importe quoi. Cependant on remarque que plus on augmente  $P_{moteurs} = 120$  plus la trajectoire s'améliore.

→ Sa trajectoire sur un demi-mètre est donc très correcte peu importe la valeur

de  $P_{moteurs}$  mais elle est particulièrement exceptionnelle pour  $P_{moteurs} = 255$

Ainsi les écarts entre le point d'arrivée pour une trajectoire parfaite et les points d'arrivée pour nos trajectoires réelles, sur une distance de 0.5 m, sont :  $e_{80} = 0.205$  m et  $e_{120} = 0.005$  m

Pour ce qui est des relevés de vitesse de déplacement, on trouve :  $V_{80} = 0.0064$  m.s<sup>-1</sup> et  $V_{255} = 0.026$  m.s<sup>-1</sup>.

### 5.3.2 Confrontation des résultats

On regroupe toutes les données que nous avons cités plus haut dans le tableau ci-dessous, ce qui facilitera grandement l'analyse de celles-ci. Cette dernière partie va donc citer les points forts et points faibles de chaque modèle ainsi que les applications concrètes dans le domaine scientifique auxquelles on destine l'utilisation de chacun des modèles (en fonction de leurs qualités et défauts).

	Vitesse Avancée ( en m/s)			Vitesse Rotation (en tr/min) P = 80	Précision Trajectoire (en m/0.5m) P = 80	Précision Trajectoire (en m/0.5m) P = 120	Précision Trajectoire (en m/0.5m) P = 255
	80	120	250				
Basique	0.008	0.03	Inf	7.5	0.250	Inf	Inf
Brosse à dent, modification	Inf	0.098	0.3	14	Inf	0.300	0.110
Brosse à dent, exosquelette	0.019	Inf	Inf	1.3	Inf	Inf	Inf
Courbé, exosquelette	0.006	0.064	Inf	3.70	0.030	0.270	Inf
Triangle Penché, exosquelette	Inf	0.085	0.18	Inf	Inf	Inf	0.325
Spider, exosquelette	0.0064	Inf	0.026	Inf	0.205	Inf	0.005

Inf = Infini (le robot fait n'importe quoi donc on décide de ne pas prendre en compte cette mesure)

FIGURE 16 – Tableau regroupant toutes les données de vitesses et de trajectoires

Premièrement on remarque que l'exosquelette "Brosse à dent" ne répond pas du tout aux exigences d'un bon exosquelette, peu importe l'utilisation que l'on veut en faire, celui-ci avance néanmoins plus vite que le Kilobot basique pour  $P_{moteur} = 80$ , mais c'est la seule chose positive qu'on peut dire sur lui. Nous pensons, du fait de la différence d'efficacité entre le Kilobot "Brosse à dent modifié" et le Kilobot équipé de l'exosquelette "Brosse à dent", que la matière joue beaucoup sur les capacités du robot. Autrement dit c'est sûrement plus le fait que les pattes soient faites d'une matière souple qui rendent le "Brosse à dent modifié" aussi efficace plutôt que le fait que ces pattes soient nombreuses et penchées.

Pour ce qui est du Kilobot "Brosse à dent modifié", on pense que c'est un très

bon élément. Il est le spécimen qui avance et tourne le plus vite, ce qui tend à vérifier notre théorie selon laquelle les pertes d'énergies dues à la possession d'un exosquelette influent vraiment beaucoup sur les capacités du robot (il va par moment jusqu'à 3 fois plus vite que le deuxième robot plus rapide de notre tableau).

Cependant le fait de ne pas posséder de corps externe ne fait pas tout, car le "Basique" n'est non plus pas muni d'un exosquelette et pourtant il fait partie des spécimens les plus lent du lot.

Pour résumer, le Kilobot "Brosse à dent modifié" :

- est rapide dans ses déplacements
- demande une puissance de vibration plus conséquente pour bien fonctionner
- et il nous donne l'information qu'il faut privilégier des matières souples .

On peut donc destiner le Kilobot "Brosse à dent modifié" à des tâches insistant sur la rapidité d'exécution, des trajets longues distances et n'impliquant pas une grande précision (du fait de sa rapidité ; il ne peut pas être très précis et rapide à la fois).

Concernant l'exosquelette "Courbé", on remarque qu'à  $P_{moteur} = 120$  il est plus rapide que le Kilobot basique, or on le rappelle, le dessein de ce prototype lorsque nous avons décidé de le créer était qu'il ait la même grande stabilité que l'exosquelette "Bloc circulaire", mais qu'il soit plus rapide que celui-ci. "Bloc circulaire" est plus lent et moins précis dans sa trajectoire que le Kilobot basique, qui lui même est plus lent et moins précis dans sa trajectoire que "Courbé", mission réussie nous avons amélioré les deux défauts principaux que nous voulions.

Pour résumer l'exosquelette "Courbé" :

- est plus rapide que "Bloc circulaire" dans ses déplacements
- est très précis dans ses déplacement
- nécessite peu de puissance de vibration pour être efficace

On peut donc destiner l'exosquelette "Courbé" à des tâches insistant sur la précision des mouvement, ne nécessitant pas une rapidité extrême, et il peut être utilisé pour des expériences ayant besoin d'éléments robustes (pour tester des algorithmes de mesures de chocs entre les Kilobots par exemple).

Pour ce qui est de l'exosquelette "Triangle penché", nous estimons qu'il est la preuve que si les pattes du Kilobot ou de son support sont penchées, le Kilobot ira forcément plus vite que si elles ne l'étaient pas. En effet pour commencer c'est tout simplement l'exosquelette le plus rapide de tous nos spécimens, et ceci peu importe la puissance à laquelle on utilise le Kilobot.

On ne lui trouve pas d'application concrète particulière pour plusieurs raisons. La première c'est que mis à part aller plus vite que ses congénères il n'a pas de grande qualité de stabilité ou précision dans sa trajectoire.

La seconde c'est tout simplement que ce prototype a été créé spécialement pour valider notre théorie sur l'influence qu'a le fait de pencher les pattes de l'exosquelette,

sur la vitesse. Il le fait, il remplit son rôle et c'est tout ce qu'on lui demande pour l'instant.

Pour finir, à haute puissance de vibration, la précision dans la trajectoire de l'exosquelette "Spider" est remarquable. On est pas loin de la ligne droite parfaite et c'est très surprenant, il surpasse tout bonnement tous les spécimens que nous avons testé au cours de notre stage (sur la précision). L'un des désavantage cependant de ce modèle est sa vitesse de déplacement, ce n'est pas très étonnant étant donné sa taille imposante et son poids. Un point en plus peut lui être attribué sur son design mais celui-ci n'est pas très significatif concernant ses performances, mais il est bon de souligner qu'il est agréable visuellement.

Pour résumer, l'exosquelette "Spider" :

- est d'une précision "chirurgicale" dans ses déplacements, sur de courtes et longues distances, à forte puissance de vibration cependant
- est un peu lent, mais compte tenu de son poids il s'en sort bien

On peut donc destiner l'exosquelette "Spider" à des tâches insistant sur la précision extrême des mouvements, ne nécessitant également pas une rapidité extrême, et de part sa taille pouvant accueillir facilement des éléments extérieurs comme des composants électroniques par exemple (on verra cela dans la prochaine partie consacrée à la centrale d'inertie) et ceci sans que la présence de cet élément ait un trop grand impact sur la dynamique du robot.

## 5.4 Bilan motricité

Notre étude sur plusieurs morphologies nous a permis de vérifier par le biais de différentes expérimentations et études quelles caractéristiques étaient optimales pour l'amélioration de la motricité de nos modules. Nous avons en sommes pu mettre en évidence le fait que les matières souples, telle que les poils de brosse à dent, étaient plus efficace et que l'amélioration de leur stabilité résidait en un simple support axiale (une patte que l'on rajoute devant et qui sert d'appui au robot). Nous confirmons par l'étude les observations macro que nous avons faite en première partie, tout en développant des mécanismes adaptés aux tests des kilobits dans un cadre limitant le nombre de robots à casser et diversifiant les socles et supports utilisables, selon les différentes utilisations que l'on souhaite en faire.

## 6 Implémentation d'une Centrale d'inertie / Inertial Measurement Unit (IMU)

Le suivi de trajectoire est une des deux tâches qui nous ont été donné pour la validation du stage. Le principe est de mettre en place un moyen de connaître la déviation qu'un Kilobot aurait par rapport à la trajectoire initiale qu'on lui imposerait (déviation due aux imperfections de la simulation dans le monde réel). A terme, l'objectif serait de pouvoir effectuer une correction en temps réel de la trajectoire du robot lorsque celui dévierait.

Une première grande étape dans ce sens est l'utilisation d'un système électronique permettant la récupération des données de positions et d'inclinaisons du robot sur un trajet d'une plus ou moins grande distance.

Etant donné que nous travaillons sur un mécanisme se déplaçant dans un environnement 2D, l'étude se limitera donc aux composantes de déplacement sur x et y, et à la composante de rotation suivant l'axe Z assendant. Nous utiliserons ensuite ces composantes afin de représenter et corriger les écarts entre théorie affichée via l'IMU et expérimentation par le biais d'études pratiques.

Il existe déjà des systèmes de localisation de robots, mais ceux-ci ne sont à vrai dire pas très précis [5]. Pour palier à ce problème nous allons implémenter le composant électronique IMU, qui est une réponse efficace et fiable aux problématiques de cartographies que l'on rencontre actuellement sur de si petits modules que sont les Kilobots.

Une IMU est un composant électronique contenant en son sein plusieurs MEMS (Microsystème électromécanique) tel qu'un accéléromètre, un magnétomètre, etc [8]. Cet outil va en fait servir de coupleur afin de combiner les données de ces différents MEMS pour gagner en précision. Par exemple, le MPU-6050, composant que nous utiliserons, est un composant faisant partie de la catégorie des IMU et qui est très utilisé pour le calcul d'accélération, pour des supports d'application 2D et 3D. Il combine en fait un gyroscope à 3 axes et un accéléromètre à 3 axes ainsi qu'un processeur de mouvement numérique intégré afin de calculer des accélérations axiales et rotationnelles [9].

Comme dit précédemment on ne va pas utiliser toutes les capacités que l'IMU offre, mais assez pour l'utilité que l'on veut en faire.

Notre but est donc de travailler sur ce système de localisation plus précise, et de trouver un moyen de l'appliquer aux Kilobots.

### 6.1 Code de localisation

Dans un premier temps nous étudions l'IMU qui nous a été fourni en la testant via un code récupérant les données angulaires du Kilobot. Nous transmettrons par la suite ces accélérations au PC à l'aide d'une connexion filaire : on cherche à afficher les valeurs de l'IMU et à observer le temps et la précision des résultats fournis par l'IMU.

Si nous limitons le code à l'affichage des accélérations angulaires, c'est parce que lors d'une étude plan (2D), la connaissance de la rotation selon l'axe Z ascendant

est suffisante pour déterminer de combien de degré le module étudié a dévié. En sommes, pour déterminer une déviation de trajectoire, il faut simplement connaître l'angle que fait l'axe du Kilobot avec l'axe de la trajectoire droite parfaite.

Pour cette étude, nous avons utilisé une carte Arduino Nano pour ce qui est du traitement de donnée (l'un des plus petit modèle de carte d'Arduino), et une centrale inertielle de type MPU-6050 pour les mesures d'accélération.

On commence donc par faire fonctionner un code qui existe déjà pour afficher sur le moniteur série Arduino les données de l'IMU que l'on utilise (qui est globalement le modèle d'IMU utilisé pour toutes les applications de ce type). Cependant nous modifions quelques peu le code pour l'appliquer à notre propre matériel, puis nous le lançons en faisant tourner l'IMU selon 3 axes. Le principe est d'identifier les données angulaires correspondant aux différents axes de notre Kilobot : l'axe de roulis, l'axe de tangage et l'axe de cap.

Nous avons ci-dessous affiché ce que l'on voit lorsqu'on lance le code dans le cas où l'IMU reste immobile. A savoir que toutes les données sont des angles en degrés, compris dans l'intervalle -180 à 180 degrés.

ypr	33.58	21.31	40.42
ypr	33.58	21.31	40.43
ypr	33.58	21.31	40.42
ypr	33.57	21.32	40.42
ypr	33.58	21.31	40.43
ypr	33.58	21.31	40.43
ypr	33.57	21.32	40.42
ypr	33.58	21.32	40.42
ypr	33.58	21.31	40.43
ypr	33.58	21.31	40.42
ypr	33.58	21.31	40.42
ypr	33.58	21.31	40.42
ypr	33.58	21.31	40.42
ypr	33.58	21.31	40.42
ypr	33.58	21.31	40.43
ypr	33.57	21.31	40.42
ypr	33.58	21.30	40.42
ypr	33.58	21.30	40.43
ypr	33.58	21.31	40.43
ypr	33.59	21.31	40.42
ypr	33.59	21.32	40.42
ypr	33.59	21.32	40.42
ypr	33.58	21.32	40.42
ypr	33.58	21.30	40.43

FIGURE 17 – Moniteur série Arduino affichant les valeurs d'angle de tangage, de roulis et de cap

Les lettres Y P R (pour Yaw Pitch et Roll) sont pour les différents axes rotationnels et leur position donne le numéro de la colonne qui affiche leur valeur.

Une fois que l'on a ces données élémentaires sur l'IMU, que nous savons qu'elle répond correctement et qu'elle prend des valeurs cohérente, nous allons l'étudier à travers un affichage 3D, pour fluidifier nos observations, et simplifier la compréhension d'un tel outil qu'est l'IMU.s

### 6.1.1 Modélisation 3D en temps réel d'un Kilobot

Une fois que nous savions déterminer les valeurs des différents axes, nous avons décidé de faire l'affichage sur ordinateur d'une modélisation 3D d'un Kilobot, qui utiliserait la rotation de l'IMU pour la retranscrire virtuellement. Le but est donc de faire un Kilobot qui s'inclinerait de tant de degré lorsque l'IMU serait incliné du même nombre de degrés, et ceci pour vérifier le bon fonctionnement du code et de l'IMU que l'on utilise.

Pour ce faire, nous utilisons un logiciel qui va de paire avec Arduino lorsqu'il s'agit de représenter virtuellement le mouvement de capteurs, c'est *Processing IDE*.

Il existe déjà des modélisations de quelques objets sur *Processing IDE* (des avions notamment) [7], mais nous avons jugé inapproprié d'utiliser ce type de modélisation qui n'est pas réaliste quand à l'objet de notre étude.

De ce fait nous avons créé de toutes pièces un Kilobot virtuel qui est la représentation globale et à l'échelle d'un Kilobot classique. On peut observer ci-dessous quelques captures d'écrans de la modélisation de celui-ci.

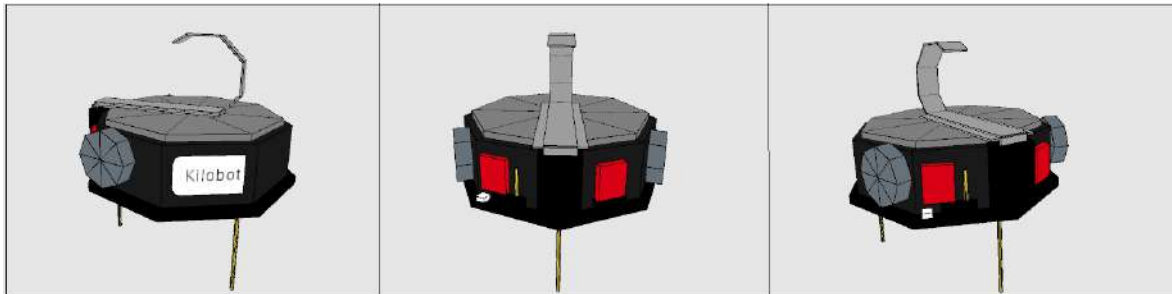


FIGURE 18 – Différentes vues du Kilobot modélisé

Une fois cet outil numérique adapté à notre recherche, nous l'avons donc testé et le bilan est positif : le modèle 3D répond aussi bien que le faisait la liste pour l'IMU et nous n'avons cette fois ci plus de barrière de langage puisque la compréhension est visuelle. Tout ceci simplifiera grandement les interactions que nous aurons avec l'IMU et nous simplifiera par la même occasion les interprétations des résultats des prochains tests.

Pour finir, étant donné le peu de références qu'il existe en terme d'objets créés via *Processing IDE* sur internet (nous n'en avons trouvé qu'une), nous réfléchissons encore à l'idée de laisser le fichier modélisant le Kilobot sur internet en open source, étant donné que le Kilobot fait l'objet de nombreuses études, il peut toujours être utile de s'assurer du bon fonctionnement de son matériel.



## 6.2 Comparaison Théorie/Pratique - Application au cas du Thymio

Pour cette partie, le but est de vérifier de manière plus poussée si l'IMU est assez performante et surtout assez fiable quand à la précision de ses résultats. Nous savons qu'elle répond bien aux mouvements lors d'une transmission filaire et que les valeurs sont cohérentes avec les rotations que l'on fait avec elle. Nous allons désormais comparer son comportement dans le cas d'une application sur un module en mouvement : le Thymio.

En effet, plutôt que de tester l'IMU directement sur le module Kilobot, nous allons l'expérimenter dans un cadre offrant plus de sécurité quand à la précision attendue des résultats. Pour ce faire nous choisissons un module bien plus imposant que le Kilobot et qui est celui du Thymio. Cette différence de taille entre les deux modules devrait s'accompagner d'une meilleure précision sur les données relevées par l'IMU.

On va par la suite implémenter l'IMU sur le Thymio et on comparera, pour une trajectoire donnée, les valeurs d'angle que nous relevons avec les valeurs d'angle que l'IMU mesure. On considèrera les valeurs que nous relevons nous même comme les angles de référence malgré le fait qu'il y ait un certain nombre d'incertitudes les concernant. Ces erreurs sont notamment liées aux procédés mis en place pour relever les valeurs expérimentales, incluant notamment une erreur humaine et matériel avec le rapporteur qui sert à mesurer l'angle.

De ce fait tous les écarts entre nos valeurs et celles de l'IMU seront considérés comme des erreurs de l'IMU.

Concernant le matériel pour faire ces tests, nous avons besoin de l'IMU et de tout ce qui lui permet d'envoyer des données à l'ordinateur (des fils plus généralement), d'un robot capable de porter l'électronique embarquée (le robot Thymio qui nous a été prêté pour l'occasion), un mètre, une équerre et du matériel de capture vidéo.

### 6.2.1 Protocole de relevé d'angles

- Pour commencer on se munit de notre système IMU et de tout ce qui le relie à l'ordinateur, et on fixe l'IMU sur le robot *Thymio* à l'aide de scotch. Il faut qu'il soit bien maintenu en place car le moindre écart pourrait fausser les résultats.
- En bougeant le robot de droite à gauche on confirme la position du roulis parmi les 3 colonnes qui affichent les données angulaires.
- On met en place le robot pour la simulation, on le met sur le zéro du mètre et parallèle à celui-ci à l'aide d'une équerre.
- On relève l'angle de roulis initial sur le moniteur série Arduino.
- On filme d'une vue de dessus le robot en plaçant une caméra ou un téléphone au dessus de celui-ci, à hauteur raisonnable (de l'ordre du mètre), bien parallèle à la surface sur laquelle est posé le robot et surtout immobile.
- On lance le robot en appuyant sur le bouton lui donnant l'instruction d'avancer tout droit.
- Une fois que le robot a atteint la position finale (nous avons choisi de faire 3 positions finales : 60 cm, 100 cm et 120 cm), on appuie sur le bouton d'arrêt

et on ne touche pas au robot.

- On arrête ensuite de filmer et une fois que les données que l'IMU nous transmet se sont stabilisées, on relève l'angle de roulis final sur le moniteur série Arduino.
- On place ces données dans un tableau et on réitère l'expérience un nombre de fois qu'on estime suffisant, pour faire une moyenne (nous avons effectué l'expérience 5 fois pour chacune des 3 distances de test ce qui nous fait 15 points en tout).

Un point sur lequel on met l'accent lors de nos mesures est de réduire au maximum les contraintes appliquées sur le robot et qui sont liées au fait qu'il est relié par un câble à l'ordinateur. C'est la source présentant le plus haut risque d'erreur.

Ci-dessous on peut trouver un schéma explicatif qui montre la disposition des éléments utilisés pour mener à bien ce protocole.

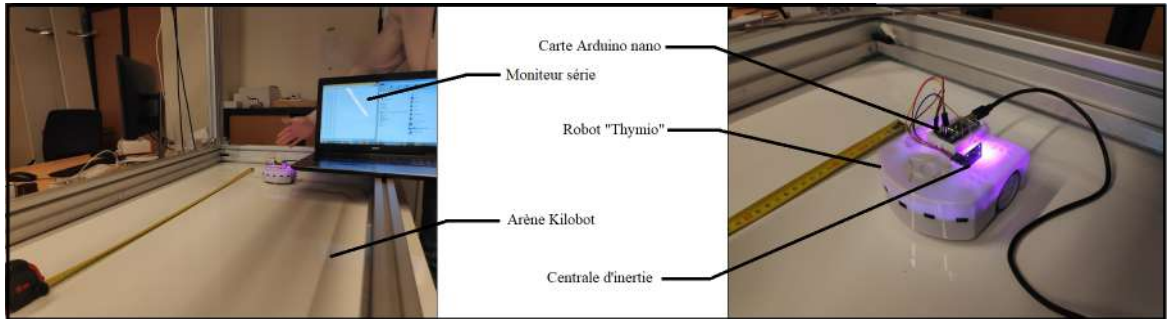


FIGURE 19 – Schéma explicatif sur la disposition des éléments de notre montage

### 6.3 Résultats

Une fois que nous avons appliqué le protocole pour les 3 distances différentes, on peut classer toutes nos données dans un tableau et en déduire un écart entre la théorie et la pratique. On applique la formule suivante pour le calcul de l'écart entre pratique et théorie (résultat de la dernière colonne) :

$$\frac{EcartroulisIMU}{Ecartroulisexp} * 100 \quad (1)$$

Ce calcul est en fait simplement le rapport entre l'angle relevé par nos soins et l'angle calculé par l'IMU, et cela nous donne une idée d'à quel point les valeurs données par l'IMU sont proches de la réalité. On note ainsi que la dernière colonne donne le "Taux de ressemblance" entre la théorie donnée par l'IMU et les données expérimentales. Par exemple, une valeur de 10 dans cette colonne signifierait que la théorie et l'expérimentation n'ont que 10 pourcent de similitude, ce qui serait très faible.

Longueur trajet (en cm)	Roulis départ IMU (en °)	Roulis arrivée IMU (en °)	Écart roulis IMU (en °)	Écarts roulis expérimental (en °)	Écart relatif IMU/expérimental (en %)
60	-4.32	-4.30	0.02	0.8	2.5
60	-4.27	-3.94	0.31	5.1	6.1
60	-4.65	-4.18	0.47	4.3	10.9
60	-4.28	-3.89	0.39	3.6	11
60	-4.35	-4.03	0.32	2.7	11.8
Moyenne	X	X	0.302	3.3	8.46
-	-	-	-	-	-
100	-1.81	-2.08	0.27	3.2	8.5
100	-2.92	-2.65	0.27	3.4	8.1
100	-2.34	-2.50	0.16	2.1	7.6
100	-3.70	-3.21	0.49	4.1	12
100	-4.02	-3.59	0.43	3.9	10.8
Moyenne	X	X	0.324	3.38	9.4
-	-	-	-	-	-
120	-3.24	-2.97	0.27	3.0	9.1
120	-4.70	-2.93	1.77	4.2	42.1
120	-4.85	-3.99	0.86	2.0	43.0
Moyenne	X	X	0.97	5.8	31.4

FIGURE 20 – Tableau contenant les angles calculés par l’IMU et les angles calculés de manières expérimentale en fonction de la distance que le robot à parcouru

La première chose qui nous apparaît en lisant le tableau c’est que l’angle calculé par l’IMU est bien différent de l’angle qu’on a relevé expérimentalement. Nous nous attendions à ce que les valeurs ne soient pas totalement les mêmes mais on remarque que l’écart est quand même de quelques degrés. Or, on s’attendait à un écart relatif en moyenne de 80 - 90 %, et au final on tourne plus autour des 10 % avec une valeur max de 43 %. Ceci dit il y a des incertitudes que nous ne maîtrisons pas, et on remarque ici que les prises de mesures avec l’IMU font état d’une difficultés plus conséquente que celle à laquelle nous nous attendions.

Toutefois, avant de faire des conjectures sur l’origine de ces résultats et des erreurs qui y sont liées, nous allons représenter de manière plus schématique les écarts sur les angles de roulis de l’IMU et les écarts sur les angles de roulis expérimentaux, en affichant ces données sur un seul et même graphique. Comme nous l’avions fait avec l’affichage 3D des accélérations du Kilobots, nous aurons ici une compréhension visuelle de notre problème, en espérant que cela nous apporte des éléments pour l’interprétation générale de nos données.

Grâce au schéma ci-dessous on va pouvoir visualiser explicitement la fidélité des données que calcule l’IMU.

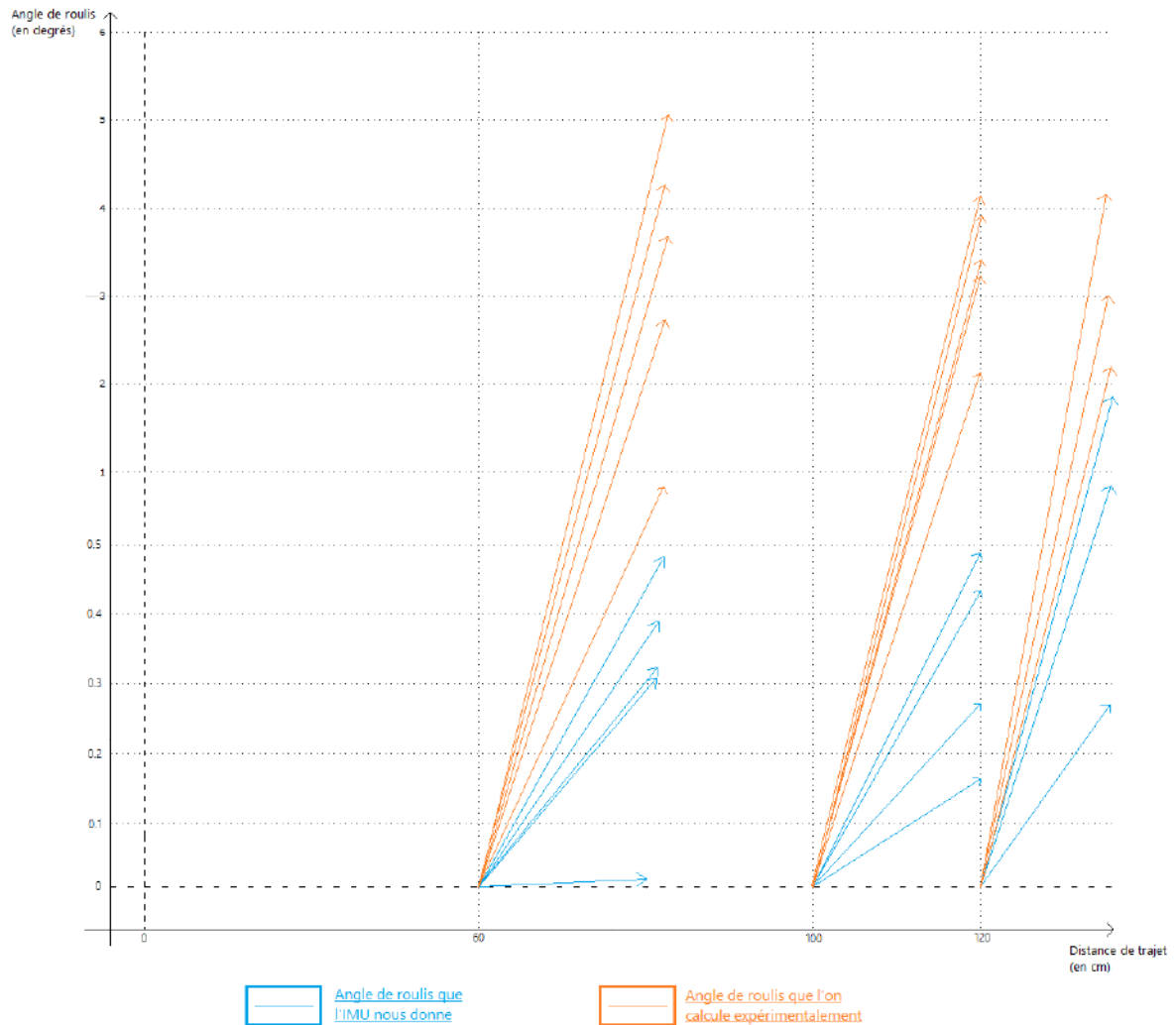


FIGURE 21 – Schéma des angles de roulis expérimentaux et de l'IMU en fonction de la distance que le robot à parcouru

Chaque flèche du schéma représente une direction. Par exemple, si on considère les flèches à 60cm, on remarque que les mesures expérimentales donne un angle plus important que celles de la théorie. On retrouve bien le comportement observé dans le tableau précédent. Ce schéma se retrouve d'ailleurs pour les 3 distances 60, 100 et 120.

Néanmoins, on remarque que les flèches ne vont globalement pas dans tout les sens et les valeurs restent dans un intervalle resserré, ce qui montre visuellement la fidélité des mesures d'angles de l'IMU.

Par ailleurs, mis à part pour la valeur d'angle  $0.02^\circ$  que l'IMU nous donne pour un trajet de 60 cm, les valeurs d'angles sont assez centrées et surtout celles-ci ont tendance à augmenter quand la durée du trajet augmente. Pour ce dernier point cela semble assez logique : si l'on considère que le robot dévie de manière constante de sa trajectoire initiale, alors plus le trajet dure longtemps plus le robot est censé former un angle élevé avec la ligne droite parfaite de sa trajectoire initiale.

## 6.4 Etude des sources d'erreur

Pour ce qui est de l'interprétation des erreurs, nous avons à ce jour 2 hypothèses possibles :

- soit les erreurs observées sont bien dûes à la précision de l'IMU, qui n'est donc pas adaptée pour nos types de mesures ;
- soit elles sont dûes à un biais lié à ces mêmes mesures.

Pour ce dernier point, soit c'est l'IMU qui est remise en cause du à une erreur dans la calibration, soit c'est le protocole de mesure qui est remis en cause.

D'après les valeurs recueillies dans le tableau précédent, outre les cas à 120cm, on remarque que les écarts pour l'IMU sont de l'ordre de 10%. De plus, en considérant le fait qu'aucune source d'incertitude n'a été négligée ni avant (élaboration du protocole), ni pendant les mesures (application du protocole), on peut aisément supposer que la source des erreurs observées est bel et bien liée à l'IMU elle même.

La source serait donc apparu lors de la calibration de celle-ci. Etant donné que nous n'avions à ce moment pas considéré que la calibration pouvait dépendre du module observé ou bien de l'IMU testé, il est possible que l'IMU soit donc faussé. Nous avons en effet pris en considération le calibrage de la MPU-6050 mais pas en compte le fait que notre MPU puisse être différente de celle théorique ou bien que son application sur le Thymio ne soit pas adaptée à cette même calibration.

Nous savons par ailleurs que le Thymio, contrairement au Kilobot, ne présente pas d'importantes vibrations lors de son fonctionnement, ce qui permet de conclure sur son très faible impacte sur la dérivation.

En sommes, on voit déjà schématiquement que l'écart entre expérimentation et théorie semble dépend d'un schéma qui se répète à 60, 100 et 120 cm. Pour résumer, il semblerait que dans le cadre de l'hypothèse la plus plausible à ce jour, celle considérant la calibration de l'IMU comme source majeure d'erreur, l'IMU fonctionne bien concernant la fidélité des mesures.

Arrivant au terme de notre stage, nous n'avons pas pu étudier en profondeur un moyen de corriger la trajectoire d'un Kilobot en temps réel grâce aux angles de roulis que nous avons recueillis précédemment. De même, nous n'avons pas pu effectuer dans des conditions protocolaire aussi satisfaisante une seconde batterie de mesure pour vérifier notre théorie sur la calibration. Cependant on peut spéculer sur la manière dont on pourrait corriger la trajectoire de notre robot et implémenter par la même occasion l'IMU directement sur un Kilobot.

## 6.5 Pistes concernant la correction de trajectoire

Premièrement, on trouve un moyen d'inclure l'IMU sur un Kilobot sans que celle-ci ait un impact trop important sur la dynamique du robot. Il faut que l'IMU soit centré sur le centre d'inertie du robot, ainsi seul le poids du Kilobot sera concerné par la présence de l'IMU (d'un point de vue force extérieures appliquées au Kilobot,

il n'y aura qu'une résultante pour la masse de l'IMU et aucun moment ce qui ne perturbera pas la stabilité de celui-ci).

Ensuite il faut trouver un moyen de faire communiquer les données de l'IMU à la base de contrôle. Or on rappelle que celle-ci ne peut recevoir des données que de façon filaire, donc un moyen certes un peu bricolé mais qui pourrait sûrement marcher, serait de connecter de manière filaire un Kilobot à la base de contrôle et d'envoyer les données de l'IMU du Kilobot qui se déplace dans l'arène, au Kilobot accroché à la base de contrôle. En effet comme précisé dans la partie donnant les fonctionnalités du Kilobot, il dispose d'un émetteur et d'un récepteur infra-rouge. Une fois que l'information de l'IMU sera transférée au Kilobot accroché, celui-ci l'envoie filairement à la base de contrôle. Nous avons entamé cette partie mais la communication des données de l'IMU est plus complexe que la théorie le laisse supposé, rendant très difficiles la communication aussi bien filaire que par ondes infrarouges.

Pour finir il faut compiler dans le micro-contrôleur un programme qui corrigeant les erreurs. Dans le cas d'une application sur des Kilobots, les sources d'erreurs seront multiples : nous aurons à la fois la calibration de l'IMU à prendre en compte mais aussi le fait que le Kilobot vibre pour avancer et que la position de l'IMU ne pourra pas être exactement au centre comme la théorie le laisse penser.

Ainsi, le programme à compiler comprendrait une calibration qui ferait vibrer le moteur du côté duquel le Kilobot dévie un peu plus fort que l'autre (si le Kilobot dévie vers la droite on augmente la puissance de vibration du moteur droit, et inversement si il dévie vers la gauche) ce qui permettra de remettre le Kilobot sur le droit chemin.

C'est là la solution que nous avons trouvée pour effectuer de la correction de trajectoire. Etant donné l'aspect "fait maison" de cette méthode, il faudrait à terme créer une base de contrôle qui serait capable de recevoir des données infra-rouge ; ainsi qu'un système qui automatiserait la pose de l'IMU.

## 7 Conclusion - Bilan Stage

Pour conclure sur ce stage et les missions qui nous incombait, nous n'avons pleinement terminé que l'une de nos deux tâches. Ainsi, l'aspect motricité et morphologique des Kilobots a été étudié en profondeur, amenant des données expérimentales et numériques, permettant de conclure sur les supports optimaux et les méthodes d'implémentation idéales sur les Kilobot.

Pour ce qui est de l'aspect IMU, il en résulte que les valeurs finales restent biaisées ; ce qui ne nous empêche pas tout de même de conclure sur nos protocoles de mesure et notre méthodologie générale de travail. En effet, ces derniers se sont avérés particulièrement efficaces, nous permettant d'établir un échelonnage dans nos études, passant de la théorie, à l'expérimentation filaire puis au cas d'un robot théoriquement plus sûr au niveau des incertitudes, et de finalement émettre des hypothèses et des pistes précises quand à l'origine de nos écarts de mesure.

## Références

- [1] Kteam *Kilobot Documentation*. <https://www.k-team.com/mobile-robotics-products/kilobot#specifications>. Page constructeur. En ligne. Consulté le 16-déc-2019.
- [2] X. Xu J.A.R. Marshall A. Reina A. Font Llenas, M.S. Talamali. Quality sensitive *Foraging by a robot swarm through virtual pheromone trails*. <https://github.com/Di0DeProject/PheromoneKilobot>, 2018. Livre. Consulté le 07-nov-2019.
- [3] Drikjan Krijnders Erik Duisterwinkel Anil Yaman, Elena Talmishnikh. subcultron *Submarine cultures perform long-term exploration of unconventional environmental niches*. <http://www.subcultron.eu/project-description/>, 2014. Site Web. Consulté le 16-déc-2019.
- [4] Drikjan Krijnders Erik Duisterwinkel Anil Yaman, Elena Talmishnikh. textitThe Phoenix Project. [www.phoenix-project.eu](http://www.phoenix-project.eu), 2014. Site Web. Consulté le 16-déc-2019.
- [5] Anthony Antoun, Gabriele Valentini, Etienne Hocquard, Bernat Wiandt, Vito Trianni, and Marco Dorigo. *Documentation Kilogrid*. <http://iridia.ulb.ac.be/kilogrid/documentation.html>, 2016. En ligne. Consulté le 16-déc-2019.
- [6] Anthony Antoun, Gabriele Valentini, Etienne Hocquard, Bernat Wiandt, Vito Trianni, and Marco Dorigo. *Kilogrid : a Modular Virtualization Environment for the Kilobot Robot*. <http://iridia.ulb.ac.be/kilogrid/files/publications/iros.pdf>, 2016. En ligne. Consulté le 16-déc-2019.
- [7] Mukherjee Aritro. Project hub - arduino *Getting started with IMU (6 DOF) motion sensor*. <https://create.arduino.cc/projecthub/Aritro/getting-started-with-imu-6-dof-motion-sensor-96e066>, Février 2016. En ligne. Consulté le 16-déc-2019.
- [8] Jonathan Bernstein. Project and Article Hub *An Overview of MEMS Inertial Sensing Technology*. <https://www.fierceelectronics.com/components/overview-mems-inertial-sensing-technology>, Février 2003. En ligne. Consulté le 16-déc-2019.
- [9] Zhang Chenj. *Documentation MPU6050*. <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>, Juin 2015. Code. En ligne. Consulté le 16-déc-2019.
- [10] ISIR Laboratory. Amac : *Architectures and Models for Adaptation and Cognition*. [http://www.isir.upmc.fr/index.php?op=view\\_equipe&id=1&lang=en](http://www.isir.upmc.fr/index.php?op=view_equipe&id=1&lang=en). Site Web. Consulté le 17-déc-2019.
- [11] ISIR Laboratory. Isir : *Institut des Systèmes Intelligents et de Robotique*. [http://www.isir.upmc.fr/index.php?op=view\\_page&id=2&lang=en&menuid=1](http://www.isir.upmc.fr/index.php?op=view_page&id=2&lang=en&menuid=1). Site Web. Consulté le 17-déc-2019.
- [12] LCM Laboratory. textitFlyind Ad-Hoc Networks. <https://smavnet.epfl.ch/>, Décembre 2013. Site Web. Consulté le 16-déc-2019.
- [13] X. Li M. López-Ibáñez K. Ohkura C. Pinciroli et T. Stützle Éd M. Dorigo, M. Birattari. Lecture notes in computer science *Swarm intelligence : 10th*

- international conference at Brussels, Belgium.* 62 :185–196, Septembre 2016. Livre. Consulté le 12-oct-2019.
- [14] Nadia Nedjah and Silva Junior Luneque. *Review of methodologies and tasks in swarm robotics towards standardization.* *Swarm and Evolutionary Computation*, 50(50) :100565, Août 2019. Received 18 October 2017, Revised 10 July 2019, Accepted 10 August 2019, Available online 15 August 2019. Livre. Consulté le 16-déc-2019.
  - [15] Caroline Perry. Harvard scholl of engineering : *A self-organizing thousand-robot swarm.* <https://www.seas.harvard.edu/news/2014/08/self-organizing-thousand-robot-swarm>, Août 2014. Site Web. Consulté le 17-déc-2019.
  - [16] Andreagiovanni Reina, Alex J. Cope, Eleftherios Nikolaidis, James A.R. Marshall, and Chelsea Sabo. *ARK : Augmented Reality for Kilobots.* [https://areina.staff.shef.ac.uk/pdf/reina\\_RAL\\_2017.pdf](https://areina.staff.shef.ac.uk/pdf/reina_RAL_2017.pdf), Avril 2017. En ligne. Consulté le 16-déc-2019.
  - [17] Michael Rubenstein, Christian Ahler, Nick Hoff, Adrian Cabrera, and Radhika Nagpal. Robotics and autonomous systems *Kilobot : A low cost robot with scalable operations designed for collective behaviors.* 62(7) :966–975, Juillet 2014. Livre. Consulté le 27-sept-2019.
  - [18] Harvard University. *Kilobot Labs.* <https://www.kilobotics.com/labs>, 2013. Documentation concepteur. En ligne. Consulté le 16-déc-2019.
  - [19] SSR Lab : Harvard University. *Documentation Kilobotics.* <https://www.kilobotics.com/documentation>, Avril 2013. Documentation constructeur. En ligne. Consulté le 16-déc-2019.