



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : RTIC9/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Réseaux et Technologies de l'Information et de la Communication (RTIC)

Conception et réalisation d'un robot mobile fennec camouflé basé sur l'IoT pour des applications militaires

Par :

BOUDJELKHA ICHRAK

Soutenu le 26/06/2022 devant le jury composé de :

AYAD Soheyb

MCA

Président

ZOUAI Meftah

MAB

Rapporteur

AMMARI Asma

MAB

Examineur

Année universitaire 2021-2022

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة محمد خيضر بسكرة

تصريح شرفي

(خاص بالالتزام بقواعد النزاهة العلمية لإنجاز بحث)

أنا الممضي أسفله،

السيد (ة): بوجلخة اشراق الصفة: طالبة

الحامل لبطاقة التعريف الوطنية رقم: 207935099 والصادرة بتاريخ: 24.05.2022

المسجل بكلية: العلوم الدقيقة وعلوم الطبيعة والحياة

قسم: الإعلام الآلي

والمكلف بإنجاز مذكرة تخرج في الماستر عنونها:

Conception et réalisation d'un robot mobile fennec basé sur l'IoT pour des applications militaires

أصرح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات المهنية والنزاهة الأكاديمية المطلوبة في إنجاز البحث المذكور أعلاه.

التاريخ: 20.06.2022.

توقيع المعني:

Remerciements

Je remercie dieu le tout puissant de ma donnée la santé et la volonté d'accomplir cette mémoire.

*Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de **Dr. Meftah ZOUAI** , je le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et surtout sa disponibilité.*

Je tiens aussi à remercier les membres du jury pour avoir accepté d'examiner et d'évaluer ce travail.

Mes salutations vont aussi destinés à tous ceux qui ont collaboré à notre formation en particulier les enseignants du département d'informatique, Université Mohamed Khider- Biskra.

Je tiens remercier aussi le chef du département pour ses efforts pour maintenir le bon ordre public et étudier dans une atmosphère appropriée.

Ichrak BOUDJELKHA

Dédicace

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chers frères, pour leur appui et leur encouragement, A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible,

Merci d'être toujours là pour moi.

Résumé

Les pays dépensent d'énormes capitaux pour sécuriser leurs frontières terrestres et aériennes contre les infiltrés (passeurs et immigrants illégaux) en construisant des casernes et en déployant des véhicules et des forces militaires. Certaines institutions de défense utilisent les robots dans le contrôle des frontières terrestres et les drones dans le contrôle des frontières aériennes. L'efficacité des robots est très élevée par rapport aux éléments humains.

Dans ce projet, nous proposons de créer un robot qui est abondant dans le désert algérien, afin d'utiliser ce robot pour surveiller les frontières sud du pays.

Le robot a essentiellement des capteurs sonores afin d'éviter les obstacles. La partie mécanique du robot est équipée de servomoteurs qui aident au mouvement du robot. Le robot peut tranquillement infiltrer la zone ennemie et envoyer des informations au dashboard. L'objectif principal de ce projet est de renforcer la défense en utilisant la robotique, ce qui aidera la défense à protéger les vies humaines et à réduire le coût du contrôle terrestre de la frontière.

Mots clés : robot, surveillance, internet des objets, capteurs sans fil.

Abstract

Countries spend huge capital in securing their land and air borders from infiltrators (smugglers and illegal immigrants) by building barracks and deploying military vehicles and forces. Some defense institutions also use robots to monitor land borders, and drones to monitor air borders, and the efficiency of robots is very high compared to human elements.

In this project, we propose to create a robot which is abundant in the Algerian desert, in order to use this robot in monitoring the southern borders of the country. The robot mainly contains sound sensors in order to avoid obstacles, and the mechanical part The robot is equipped with servo motors that help in the movement of the robot, The robot can quietly infiltrate the enemy area and send information to the dashboard. The main thrust of this project is to make defense stronger by using robotics, which will help the defense protect human lives and reduce the cost of land control of the border.

Keywords : robot, monitoring, internet of things, wireless sensors network.

ملخص

تنفق الدول رؤوس أموال ضخمة في تأمين حدودها البرية والجوية من المخترقين (المهربين والمهاجرين الغير شرعيين) عبر بناء الثكنات ونشر أليات وقوات عسكرية. كما تستخدم بعض المؤسسات الدفاعية الروبوتات في مراقبة الحدود البرية والطائرات بدون طيار في مراقبة الحدود الجوية وتكون كفاءة الروبوتات عالية جدًا مقارنة العناصر البشرية.

يعتبر التمويه تقنية عسكرية ذات دورا هاما في حماية الأفراد (الجنود والعساكر) وأجهزة المراقبة من الاعتداء عليها أو إتلافها من طرف الأعداء، حيث يتم صنع بدلات للعسكريين أو صناعة روبوتات تحاكي في شكلها الألوان والحيوانات والأشجار الموجودة في البيئة المراد مراقبتها، مما تسمح هذه التقنية بخداع الأعداء ولا تسمح لهم بالتفطن لوجود أنظمة حماية بشرية كانت أو تقنية، مما يحقق عنصر مفاجئة العدو.

في هذه المشروع نقترح إنشاء روبوت مموه في شكل ثعلب الصحراء (الفنك) الموجود بكثرة في الصحراء الجزائرية من أجل استخدام هذا الروبوت في مراقبة الحدود الجنوبية للبلاد، يحتوي الروبوت بشكل أساسي على كاميرا، والتي تلتقط الصور وحساسات صوتية من أجل تفادي العوائق كما أن الجزء الميكانيكي للروبوت مزود بمحركات مؤازرة تساعد في حركة الروبوت، كجزء من وظيفة التمويه يمكن للروبوت التسلل بهدوء إلى منطقة العدو وإرسال المعلومات عبر الكاميرا إلى وحدة التحكم. الدافع الرئيسي لهذه المشروع هو جعل الدفاع أقوى باستخدام الروبوتات، والتي ستساعد الدفاع على حماية الأرواح البشرية وتقليل كلفة المراقبة البرية للحدود.

الكلمات المفتاحية: روبوت، مراقبة، أنترانت الأشياء، أجهزة استشعار لاسلكية.

Table des matières

Résumé	iii
Abstract	iv
Introduction Générale	1
I Généralités sur les robots	3
I.1 Introduction	3
I.2 Définition d'un robot	4
I.3 Principe de fonctionnement de robot	4
I.4 Caractéristiques d'un robot	5
I.5 Types des robots	5
I.5.1 Robots manipulateurs	6
I.5.2 Robots mobiles	6
I.5.3 Types des robots mobiles	7
I.5.3.1 Robot mobiles à roues	7
I.5.3.2 Robots mobiles à chenilles	8
I.5.3.3 Robots volants	9
I.5.3.4 Robots marcheurs	9
I.5.3.4.1 Monopèdes	10

I.5.3.4.2	Bipèdes	10
I.5.3.4.3	Robots quadrupèdes	11
I.5.3.4.4	Robots hexapodes	12
I.6	Domaines d’applications des robots	13
I.7	Avantages des robots	14
I.8	Inconvénients des robots	15
I.9	Conclusion	15
II	Internet des objets	17
II.1	Introduction	17
II.2	Définition d’IoT	17
II.3	Historique	18
II.4	Domaines d’applications de l’IoT	18
II.5	Fonctionnement de l’IoT	19
II.6	Importance d’IoT	20
II.7	Composants d’un système IoT	20
II.8	Technologies qui ont rendu l’IoT possible	21
II.9	Objet connecté	21
II.9.1	Avantages des objets connectés	22
II.9.2	Inconvénients des objets connectés	22
II.9.3	Communication Capteur/Cloud	23
II.9.3.1	Généralités sur les capteurs	23
II.9.3.1.1	Définition d’un capteur	24
II.9.3.1.2	Réseaux de capteurs sans fils	24
II.9.3.1.3	Caractéristiques des réseaux de capteurs	25
II.9.3.1.4	Types des nœuds capteurs	25

II.9.3.1.5	Domaines d'applications des réseaux de capteurs sans fils	26
II.9.3.1.6	Classifications des capteurs	27
II.9.3.1.6.1	Types de capteur	27
II.9.3.1.6.2	Type de signal produit par le capteur	28
II.9.3.1.6.3	Type d'appareil de mesure	28
II.9.3.2	Choix de capteurs	29
II.9.3.2.1	Durabilité	29
II.9.3.2.2	Précision	30
II.9.3.2.3	Polyvalence	30
II.9.3.2.4	Consommation d'énergie	30
II.9.3.2.5	Considérations environnementales particulières	31
II.9.3.2.6	Coût	31
II.9.3.3	Inconvénients et contraintes des WSN	31
II.9.4	Définition d'un dispositif	32
II.9.4.1	Types d'informations	32
II.9.4.1.1	Téléométrie	33
II.9.4.1.2	Métadonnées de l'appareil	33
II.9.4.1.3	Commandes de périphérique	33
II.9.4.1.4	Informations opérationnelles	34
II.9.5	Motivation sur le Cloud dans l'IoT	34
II.9.5.1	Cloud Computing	34
II.9.5.2	Avantages du Cloud	35
II.9.5.3	Limites du Cloud	35
II.10	Protocoles de communication	35

II.10.1 Protocole MQTT	36
II.10.1.1 Principe de fonctionnement du MQTT	36
II.10.2 Protocole HTTP	37
II.10.3 Comparaison des fonctionnalités générales des protocoles . . .	37
II.11 Travaux connexes	38
II.12 Conclusion	39
III Conception du système	40
III.1 Introduction	40
III.2 Architecture générale du système	40
III.3 Architecture détaillée du robot	41
III.3.1 Module de déplacement	42
III.3.1.1 Mécanisme du Leg	43
III.3.1.1.1 Modèle mathématique de pas	43
III.3.1.2 Algorithmes de déplacement	44
III.3.1.2.1 Algorithme de leg	44
III.3.1.2.2 Algorithme synchronisation des legs	44
III.3.1.3 Synchronisations entre les legs du robot	45
III.3.2 Module de perception	46
III.3.3 Module de traitement	47
III.3.4 Module de communication	47
III.3.5 Module d'énergie	47
III.4 Diagramme UML	47
III.4.1 Diagramme de fonctionnement	48
III.4.2 Diagramme de classe	49
III.4.3 Diagramme de séquence	52

III.5 Conclusion	53
IV Implémentation du système	54
IV.1 Introduction	54
IV.2 Environnement de développement	54
IV.2.1 Langages de programmation et Framework	55
IV.2.1.1 Python	55
IV.2.1.2 Java	55
IV.2.2 Outils de développement	56
IV.2.2.1 Netbeans	56
IV.2.2.1.1 JDK	56
IV.2.2.1.2 JVM	56
IV.2.2.2 Visual Studio Code	57
IV.3 Mécanisme du Leg	57
IV.4 Résultat	62
IV.4.1 Simulation	62
IV.4.2 Réalisation	70
IV.4.2.1 Schéma électrique	70
IV.4.2.1.1 Servomoteur	71
IV.4.2.1.2 Capteur ultrason	72
IV.4.2.1.3 Raspberry pi	73
IV.4.2.1.4 PCA9685	73
IV.4.2.2 Présentation du matériel dans le robot	74
IV.5 Conclusion	75
Conlusion Générale	77

Bibliographies

78

Table des figures

I.1	Principe de fonctionnement général de robot [12]	4
I.2	Différents types des robots mobiles[34]	7
I.3	Robots mobiles à roues[34].	8
I.4	Robots mobiles à chenilles[34].	8
I.5	Robots mobiles volants [34].	9
I.6	Robots marcheurs[30]	10
I.7	Robots marcheurs monopèdes[30]	10
I.8	Robots marcheurs bipèdes[30]	11
I.9	Robots marcheurs quadrepèdes[30]	12
I.10	Robots marcheurs hexapodes[30]	13
II.1	Evolution de l’IoT [31].	18
II.2	Architecturs générale d’IoT[5].	20
II.3	Communication Capteur/Cloud[1]	23
II.4	Architecture du WSN[9].	25
II.5	Type des données (Télémetries) produisent par IoT [1]	33
II.6	Architecture du protocole MQTT[1].	36
II.7	Architecture du protocole HTTP[1].	37

III.1 Architecture générale du système.	41
III.2 Arcitecture de robot.	42
III.3 Modèle de leg implémenté.	43
III.4 Synchronisation entre les legs.	46
III.5 Diagramme de foctionnement de robot.	49
III.6 Diagramme de classe.	50
III.7 Diagramme de séquence.	52
IV.1 Logo python.	55
IV.2 Logo java.	55
IV.3 Logo Netbeans.	56
IV.4 Logo Vs code.	57
IV.5 Résultat de simulation de leg en python.	62
IV.6 Ensemble des valeurs extraites du fichier csv.	63
IV.7 Traçage des courbes.	64
IV.8 Simulation des quatre legs en java.	65
IV.9 Affichage de résultat.	66
IV.10Simulation des quatre legs en java.	67
IV.11Résulat de simualtion.	68
IV.12Simulation des quatre legs en java.	69
IV.13Résulat final de simualtion d'un pas complet.	70
IV.14Schéma électrique.	71
IV.15Servomoteur et ses composants.	72
IV.16Capteur ultrasonique.	73
IV.17Raspberry pi.	73
IV.18PCA9685.	74

IV.19 Composants de robot.	75
IV.20 Résultat de simulation python.	75

Introduction Générale

La réalisation d'un système robotique mobile intelligent est liée directement à la puissance d'apprendre ou de s'adapter à son environnement, donc il est nécessaire d'utiliser des capteurs(WSN) qui fournissent la perception requise de l'environnement pour une prise de décision intelligente.

Afin de garantir la fiabilité et la disponibilité de robot,on est obligé d'utiliser l'IoT qui est considéré parmi les meilleurs et nouvelles technologies qui permettent la connectivité des objets à tous moment.

Problématique

Vu que de la grande superficie de l'Algérie, estimée a 2.381.741 km, et de ses frontières terrestres, estimées à 6385 km, qui ont rendu obligatoire le contrôle des frontières avec une technique inhabitée (sans la présence de l'être humain) à travers des robots pour réduire le déploiement de soldats et protéger le pays.

Les robots jouent un rôle important dans la protection des frontières, en particulier les robots camouflés, difficiles à détecter pour l'ennemi. Le camouflage devient une compétence de survie pour le robot et une technique défensive en même temps.

Objectif de ce travail

Nous allons concevoir un robot camouflé sous la forme de l'animal Fennec (Desert Fox), qui est largement répandu dans le désert algérien, basé sur l'Internet des objets et alimenté par l'énergie solaire, qui est une énergie propre et renouvelable disponible dans l'environnement désertique en lequel le robot vivra.

Après cette introduction générale, le mémoire est structuré comme suit :

Chapitre 1 : Généralités sur les robots Ce chapitre contient des définitions et les différents types des robots et ses classifications.

Chapitre 2 : IoT (Internet of things) Dans ce chapitre on va expliquer l'importance, le principe de fonctionnement et les protocoles de cette technologie dans notre projet.

Chapitre 3 : Conception du système Ce chapitre décrit le système en profondeur, les phases de conception du système, les outils d'implémentations, et détaille le code.

Chapitre 4 : Implémentation du système Ce chapitre présente le résultat de notre travail, explique comment des paramètres spécifiques influencent les résultats obtenus et fournit une section comparative pour montrer les nouveautés de ce travail par rapport aux précédents, test de ce projet en scénario réel.

Conclusion Générale Dans cette dernière partie , nous résumons et passons en revue nos idées et des résultats et donnant quelques perspectives.

Chapitre I

Généralités sur les robots

I.1 Introduction

Avec le progrès de la vie et l'augmentation des besoins humains, il est parfois difficile d'accomplir plusieurs tâches par les êtres humains du fait de leurs pénibilités et dangers.

Afin de pallier ce manque l'homme a été obligé de trouver une solution alternative qui est "*le robot, c'est un dispositif mécatronique (alliant mécanique, électronique et informatique)*" [11], qui a la capacité de le remplacer dans certains cas comme l'accès aux endroits inaccessibles, surveillance, fabrication, en utilisant plusieurs techniques.

Les robots (formes, structures et fonctionnalités) doivent s'adapter à l'environnement avec lequel ils interagissent, ils sont destinés aux plusieurs domaines tels que industriels, militaires, santé, scientifiques, nettoyage, agricole... , etc.

Ils ont classé en deux grandes familles : des robots à base fixe (Bras manipulateurs) et des robots mobiles.

I.2 Définition d'un robot

”*L’origine de mot robot vient du tchèque robota qui signifie travail*” [36], on peut le considérer comme un esclave, c’est un système alimenté par une source d’énergie, il se compose d’un ensemble de capteurs, un microcontrôleur et des moteurs, il a la capacité d’évoluer, et de s’adapter avec l’environnement dans lequel il interagit[36].

I.3 Principe de fonctionnement de robot

C’est une machine équipée de capacité de perception à travers des capteurs c’est-à-dire relever des mesures ou capter des informations (tension, température, etc.), les envoient au gestionnaire des tâches pour les traiter et les collecter pour prendre des décisions et appliquer les actions correspondantes, qui le permet d’agir d’une manière autonome dans son environnement en fonction de la perception assuré par leurs capteurs[28].

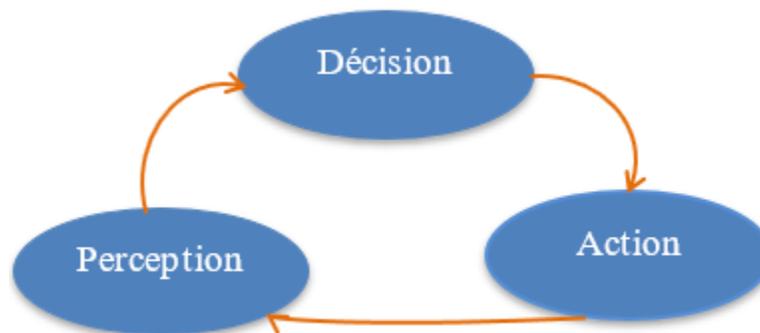


FIGURE I.1 – Principe de fonctionnement général de robot [12]

I.4 Caractéristiques d'un robot

On a cité quelques paramètres à prendre en compte dans le choix de robot[18] :

- **La charge maximale transportable** : de quelques kilos à quelques tonnes, ça dépend la rigidité de la structure et la tâche à réaliser.
- **L'espace de travail** : on s'intéresse au volume de l'espace que le robot peut atteindre via toutes les orientations possibles.
- **Le positionnement absolu** : défini par une position et une orientation dans l'espace cartésien, et le point atteint et calculé via cinématique inverse dérivée du modèle géométrique du robot.
- **La vitesse de déplacement** : pour que le robot puisse éviter les obstacles, on doit réduire la vitesse de déplacement pour qu'il est le temps de faire les mesures, les calculs et donner les ordres comme marcher ou s'arrêter.
- **Le coût** : ça dépend la mission qui va l'accomplir, mais généralement le prix est élevé en raison des services qu'il fournit.

I.5 Types des robots

En robotique il existe deux catégories :

- **Les robots à base fixe** (Les bras manipulateurs), leur importance réside dans leur stabilité.
- **Les robots mobiles**, ayant comme fonction la mobilité, (nous allons intéresser aux robots mobiles marcheurs).

I.5.1 Robots manipulateurs

A l'aide d'un bras robotisé on peut manipuler des matières très dangereuses sans contact direct dans des endroits inaccessibles pour faire des études sensibles et des développements en minimisant les risques. Leur structure en forme bras, il facilite à l'homme beaucoup d'applications dans différents domaines tels que le transport (charge lourde), la médecine (chirurgie robotique), industrie nucléaire (nucléaire, surveillance, manipulation des matériaux radioactifs), et civil dans (la lutte contre les incendies)

I.5.2 Robots mobiles

Les robots mobiles ont une place particulière en robotique à cause de ses caractéristiques. L'aspect particulier c'est la mobilité qui est le grand problème des robots manipulateurs, et pour résoudre ce problème plusieurs améliorations au niveau technologique (capteurs, énergies...), utilisation des techniques de l'intelligence artificielle afin d'accomplir des tâches plus complexes et d'ouvrir des applications dans des différents domaines, on peut dire que les robots mobiles sont réalisés afin d'accomplir le manque des bras manipulateurs. Leurs architectures dépendent des missions à effectuer et l'environnement dans lequel ils s'interagissent[34].



FIGURE I.2 – Différents types des robots mobiles[34]

I.5.3 Types des robots mobiles

Les robots mobiles sont nombreux selon leur fonction, dans cette partie on va citer et détailler quelques types pour mieux comprendre la mobilité.

I.5.3.1 Robot mobiles à roues

Les mobiles à roues sont parmi les robots les plus utilisés, appliqués et réalisés à cause de ses simples structures et formes qui se dépendent selon le nombre de roues, conception et fonction. Ils se déplacent dans toutes les directions avec une certaine accélération et vitesse importante grâce aux dimensions des roues et un agencement [40][41].

Ils ont la capacité de dépasser les obstacles (monter les escaliers) et de changer les directions dans certains cas en utilisant les nouvelles technologies[34].



FIGURE I.3 – Robots mobiles à roues[34].

I.5.3.2 Robots mobiles à chenilles

Lorsque le sol est accidenté ou d'une mauvaise qualité, les roues sont confrontées à de nombreux problèmes tels que perte d'efficacité et interaction avec le terrain qu'ils s'interagissent, et afin de les éviter, les chenilles sont la solution pour protéger les robots à roues afin d'accomplir les tâches pour lesquels ont été réalisées parce que leur utilisation présente une bonne adhérence au sol et une faculté de franchir les obstacles. Les chenilles couvrent les roues, le robot s'avance grâce eux, vaut mieux utiliser une matière de bonne qualité pour éviter le glissement et dérapage des véhicules à chenilles[34].

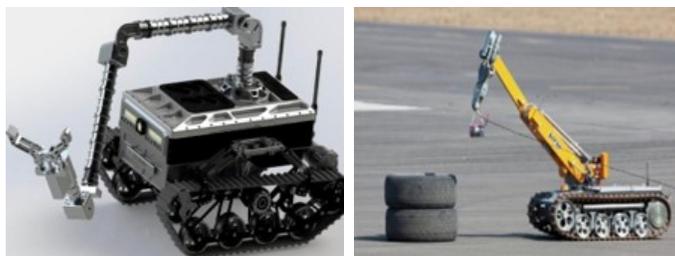


FIGURE I.4 – Robots mobiles à chenilles[34].

I.5.3.3 Robots volants

Robot volant ou un drone c'est un aéronef ou engin volant sans pilote et passagers, piloté à distance, ce genre de robots nécessite le respect de certaines règles. Il se caractérise par son apport d'information après son analyse à travers des différents capteurs, son autonomie et sa rapidité. Ils sont utilisables dans différents domaines militaires dans la lutte contre les insurrections et le terrorisme, civil dont la lutte contre les incendies, en agriculture pour la surveillance de cultures par exemple détection des plantes manquantes dans la récolte dans des zones non peuplées [34].



FIGURE I.5 – Robots mobiles volants [34].

I.5.3.4 Robots marcheurs

Lorsque l'accès à un site devient difficile et dangereux pour les humains, les marcheurs sont réalisés pour accomplir cette difficulté. Les robots mobiles marcheurs sont nombreux et très différents par leur forme et par leur nombre de pattes, à partir de ce nombre on peut distinguer les monopèdes, à deux jambes ce sont les bipèdes (humanoïdes), à quatre pattes quadrupèdes de type cheval, et à six pattes hexapodes (type araignée), octopodes ou à plusieurs pattes[30].



FIGURE I.6 – Robots marcheurs[30]

I.5.3.4.1 Monopèdes Ce type de robot à une seule jambe ou un seul pied, il saute pour se déplacer à cause de son unique point de contact avec le sol, et pour équilibrer le centre de gravité et la masse corporelle on s'intéresse à la stabilité dynamique qui est le point le plus important.

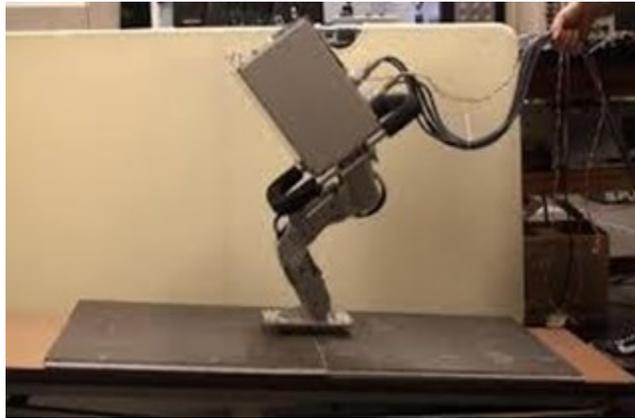


FIGURE I.7 – Robots marcheurs monopèdes[30]

I.5.3.4.2 Bipèdes Les bipèdes ou les humanoïdes, à cause leur structure semblable au corps humain qui marche sur deux pieds, ils ne sont pas stables c'est pour ça on rencontre un peu de difficulté lors l'étude de ce genre de robot. Ils peuvent interagir et s'adapter avec l'environnement, ils sont sensibles aux impacts et couples

excessifs. Les chercheurs rencontrent beaucoup de difficultés lors de la réalisation des bipèdes en raison de de la complexité des modèles mathématiques, le coût des calculs qui est difficile pour la machine et la matière de fabrication. On peut les trouver dans différents terrains, dans les aéroports pour guider les voyageurs, aussi dans les entreprises on les trouve à l'accueil, en sport dont la fédération RobotCup gère les matchs et les compétitions dont les robots sont des athlètes, chaque équipe se compose de cinq joueurs robots, ils ont comme objectif de gagner la partie.



FIGURE I.8 – Robots marcheurs bipèdes[30]

I.5.3.4.3 Robots quadrupèdes Leur structure est inspirée aux animaux de quatre pattes de type cheval, un quadrupède se déplace facilement grâce à ses pattes qui marchent séparément l'un de l'autre, ce genre est plus stable que les bipèdes, ils sont disponibles pour des différents usages soit commercial, militaire pour la surveillance des frontières ou dans les guerres ils peuvent aider les soldats et les citoyens et à usages domicile pour le nettoyage. Pour réaliser ce genre de robots on se base sur trois concepts essentiels :

- **La mobilité** : son déplacement sur tout type de sol soit régulier ou non, structuré ou non structuré.

- **Degré d'intelligence** : son degré d'autonomie, sa capacité d'adaptation et d'apprentissage.
- **Energie** : le choix de source d'énergie est très important pour garantir l'alimentation électrique de la carte mère.



FIGURE I.9 – Robots marcheurs quadrepèdes[30]

I.5.3.4.4 Robots hexapodes C'est un robot mobile marcheur sur six pattes, leur structure est inspirée des insectes (la fourmi, cigale, abeille) dont le mouvement est fondé sur trois paires de pattes, ils sont considérés plus stables que les bipèdes et quadrupèdes, cet équilibrage a été réalisé afin de minimiser la consommation énergétique. On s'intéresse à étudier ce genre de robot pour proposer une alternative des roues, ils ont une grande vitesse pour se déplacer. Ils sont utilisés dans différentes applications : domestique, il a donc l'air élégant et soigné, collection des données et échantillons dans les forêts éloignées.



FIGURE I.10 – Robots marcheurs hexapodes[30]

I.6 Domaines d’applications des robots

Les robots mobiles sont maintenant couramment utilisés pour automatiser de nombreuses tâches telles que le nettoyage, le transport, l’agriculture, la lutte contre les incendies, l’assistance aux personnes handicapées, la surveillance des frontières[18].

TABLE I.1 – Les domaines d’applications des robots[18]

Domaines d’applications	Applications
Industrie nucléaire	- Surveillance des sites. - Manipulation de matériaux radioactifs.
Sécurité civile	- Neutralisation d’activités terroristes. - Surveillance de munitions.
Militaire	- Surveillance, patrouille. - Pose d’explosifs. - Manipulation et contrôle de munitions.
Lutte contre l’incendie	- Localisation d’une source d’incendie. - Détection de fumée. - Suppression de flammes.
Nettoyage	- Coque de navire. - Nettoyage industriel.
Agricole	- Cueillette de fruits. - Traite, moisson, traitement des vignes...
Médecine	- Assistance d’urgence. - Aide aux handicapés physiques, aux aveugles.

I.7 Avantages des robots

- Augmenter le pourcentage de production.
- Réduire les coûts sur la chaîne de production.
- Améliorer la qualité des produits.
- Évitement de risque pour les opérateurs.

- Il peut travailler 24 heures d'affilée sans avoir besoin de faire de pause ni de se fatiguer.
- Après les frais d'achat, il ne nécessite aucun salaire, uniquement des frais d'entretien.
- Aucun impôt supplémentaire n'est payé pour leur service[3].

I.8 Inconvénients des robots

- L'ensemble des problèmes particuliers liés à la conception de tels robots sont :
- Le plus gros inconvénient est le prix qui est généralement élevé.
- Nécessite une maintenance préventive pour prolonger sa durée de vie, ce qui implique une dépense économique et un arrêt de la fabrication.
- Pour répondre aux besoins des robots et garantir son fonctionnement, il est nécessaire de former du personnel.
- La conception mécanique liée à la mobilité.
- La détermination de la position et de la latitude (orientation).
- La détermination du chemin optimal pour atteindre le lieu de la tâche[3].

I.9 Conclusion

Les robots se trouvent partout : dans les usines, les maisons et les hôpitaux, et même dans l'espace. Une grande partie de la recherche et du développement est investie dans le développement de robots qui interagissent directement avec les humains. Les robots sont utilisés dans les écoles afin d'augmenter la motivation des élèves à étudier les STEM (science, technologie, ingénierie et mathématiques) et comme outil pédagogique pour enseigner STEM dans un environnement concret. L'objectif de

ce chapitre est la définition de robot, citation de leur caractéristiques, ses différents types, leur classifications et leur domaines d'applications, ses avantages et ses inconvénients.

Chapitre II

Internet des objets

II.1 Introduction

Le monde a connu de nombreux développements et surtout l'internet qui a évolué d'une manière spectaculaire et incroyable dont la dernière évolution nommée IoT abréviation d'Internet des Objets (en anglais Internet of Things) qui permet la communication avec des objets ou entre eux.

Un « objet » dans « Internet des objets » est une unité de traitement capable de se connecter à Internet et d'échanger des données avec le Cloud. Les appareils sont souvent appelés "appareils intelligents" ou "appareils connectés". Ils communiquent deux types de données : la télémétrie et l'état[1].

II.2 Définition d'IoT

Internet des objets (IoT) est un vaste ensemble de technologies et de cas d'utilisation qui n'a pas de définition claire et unique, une étude a défini l'IoT comme

une technologie permet aux personnes et aux objets d'être connectés à tout moment, n'importe où, avec n'importe quoi et n'importe qui, en utilisant n'importe quel chemin/réseau et n'importe quel service[33].

II.3 Historique

L'attention portée à l'IoT augmente de façon exponentielle. En fait, le terme "Internet des objets" n'a été créé qu'en 1999. Depuis lors, le domaine de l'IoT s'est énormément développé. L'évolution de l'IoT est rapide : depuis 2014, le nombre d'objets connectés est supérieur au nombre d'humains connectés, En 2018, le nombre d'appareils IoT installés était estimé à environ 1,2 milliard, 50 milliards d'objets connectés en 2020, d'ici 2030, ce nombre devrait atteindre 125 milliards[1].

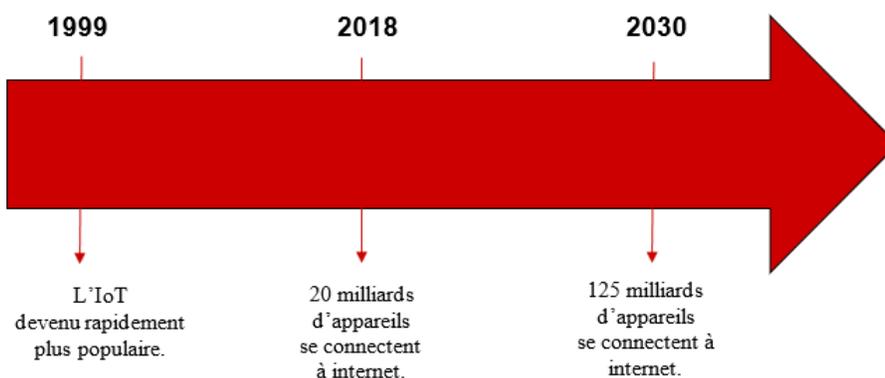


FIGURE II.1 – Evolution de l'IoT [31].

II.4 Domaines d'applications de l'IoT

Selon IHS Market, les secteurs de marché qui tirent cette croissance sont [8] :

- **Maisons intelligentes** : utilisation d'appareils intelligents pour contrôler l'environnement d'une maison.
- **Villes intelligentes** : une ville qui utilise la technologie pour améliorer l'efficacité, la durabilité et la qualité de vie des personnes vivant et travaillant dans la ville.
- **Industriel** : utilise l'apprentissage automatique et le Big Data pour générer de la valeur à partir des données des capteurs.
- **Santé connectée** : utilisation des technologies grand public pour connecter les patients et les prestataires de soins de santé en dehors de l'hôpital.

II.5 Fonctionnement de l'IoT

L'architecture d'un système IoT est identique, elle est composée de six niveaux et trois couches (couche perception, couche réseau et couche application) qui communiquent entre eux pour relier le monde des objets au monde virtuel des réseaux et du Cloud. L'internet des objets permet l'interconnexion entre les objets, un capteur commence par relever une mesure (on peut les appeler des grandeurs physiques) qui peut être, par exemple, la température, distance, humidité et la géolocalisation. Toutes les informations captées par les capteurs sont ensuite envoyées au nœud puits ou la passerelle (dit Sink en anglais) grâce à un protocole de communication de courte portée (Zigbee, Z-Wave, Wifi, BLE. . .). La passerelle à son rôle transmet ces données reçus au gestionnaire des tâches par internet ou satellite pour les analyser et prendre des décisions[27].

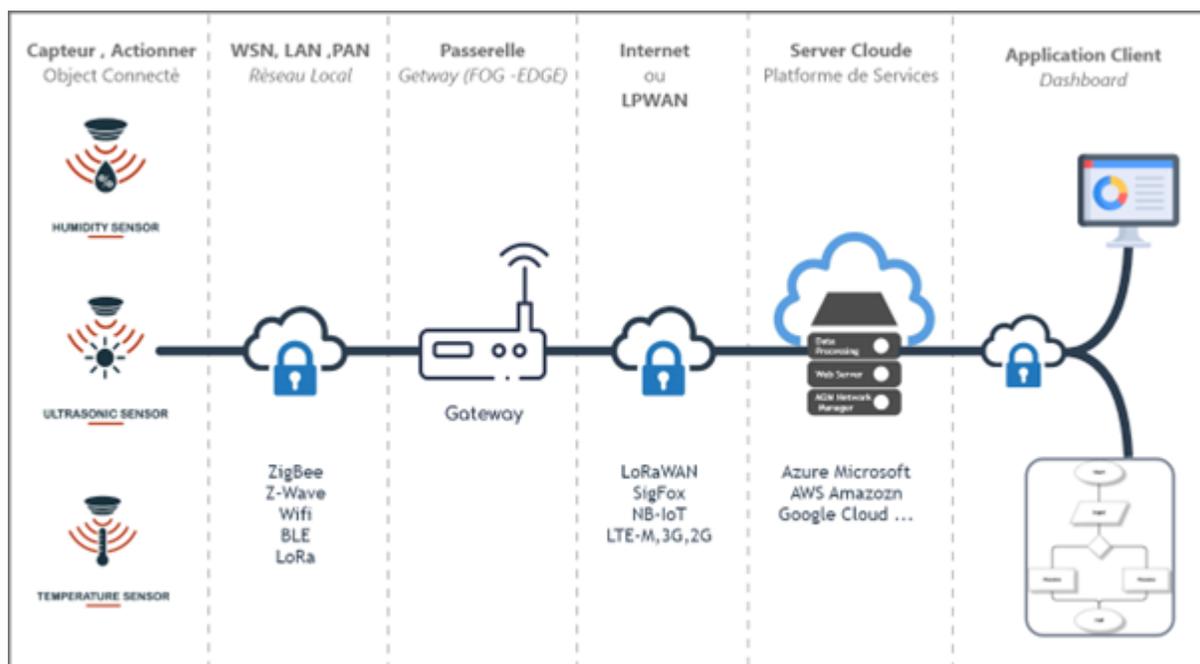


FIGURE II.2 – Architecturs générale d'IoT[5].

II.6 Importance d'IoT

- Améliorer la capacité de partage de données et faciliter la vie.
- Plus de données utiles, de meilleur décision
- Surveiller et suivre les métriques.
- Qualité de vie.
- Meilleur efficacité[16].

II.7 Composants d'un système IoT

- Des objets connectés, tels que capteurs.

- Des réseaux de communication sans fil (WSN).
- Des outils pour traitement et collecte des données, tel que le serveur Cloud.
- Des applications et services afin d’interagir avec son système connecté[1].

II.8 Technologies qui ont rendu l’IoT possible

- **Accès à une technologie de capteurs à coût réduit et faible consommation** : utilisation des capteurs fiables et moins chers pour rendre l’IoT possible.
- **Connectivité** : l’internet ou satellite a facilité la connexion des capteurs avec le Cloud et avec d’autres objets pour un transfert des données efficaces.
- **Plates-formes Cloud** : le Cloud est parmi les technologies les plus utilisées pour un IoT grâce à ses avantages dont il peut collecter un grand nombre de données, fiable car il permet aux entreprises et consommateurs d’accéder aux infrastructures.
- **Machine Learning et analyses** : sont des machines puissantes qui garantir un service et accès rapide aux serveurs Cloud pour récupérer les données plus rapidement et facilement[1].

II.9 Objet connecté

Les objets connectés sont connectés à Internet (on parle d’Internet des Objets, ou Web 3.0) : et peuvent ainsi communiquer avec d’autres systèmes pour obtenir ou fournir des informations telles que des données marketing. Cela a été rendu possible par la forte miniaturisation des composants électroniques, mais aussi par l’émergence de nouveaux réseaux de communication de type M2M. Ils peuvent par exemple :

- Collecter et stocker des informations en fonction de son environnement : fréquence cardiaque de l'utilisateur, humidité de la voûte, etc.
- Lancez une action basée sur les informations recueillies sur le web, comme arroser la pelouse à la veille d'une journée très sèche[20].

II.9.1 Avantages des objets connectés

- Taille réduite puisqu'ils ne sont pas autonomes.
- Coût intéressant en raison du format réduit et de la quantité minimisée de matériel embarqué.
- Augmentation de la sécurité et du confort du logement : la maison, les appareils électroménagers, les enceintes, les systèmes de sécurité, absolument tout peut être connecté et géré à distance.
- Valeur médicale et sanitaire : certaines des choses connectées permettent de mieux comprendre des maladies spécifiques, ce qui permet de développer des traitements plus efficaces.

II.9.2 Inconvénients des objets connectés

- Sécurité des données personnelles : Du fait qu'elles sont stockées sur Internet, il est difficile d'en garantir la confidentialité.
- Dépendance au service : si le fabricant choisit d'arrêter le service, l'objet devient inutile ou du moins n'ajoute plus de valeur.
- Dépendance à la technologie : les objets connectés s'immiscent de plus en plus dans notre quotidien, si bien que nous courons le risque de ne plus pouvoir nous en passer.
- Impact environnemental : les objets connectés sont alimentés en électricité et

sont fabriqués avec des métaux de plus en plus rares.

II.9.3 Communication Capteur/Cloud

Un capteur est un module qui observe les changements dans son environnement et envoie des informations sur ces changements à un appareil. Les appareils collectent les données des capteurs et les envoient dans le Cloud. Les appareils peuvent être très petits et avoir très peu de ressources en termes de calcul, de stockage, etc. Ils pourraient être en mesure de communiquer uniquement via des réseaux qui ne peuvent pas atteindre directement une plate-forme Cloud, comme Bluetooth Low Energy (BLE). Les appareils standard sont plus susceptibles de ressembler à de petits ordinateurs et peuvent avoir la capacité de stocker, de traiter et d'analyser des données avant de les envoyer dans le Cloud[1].

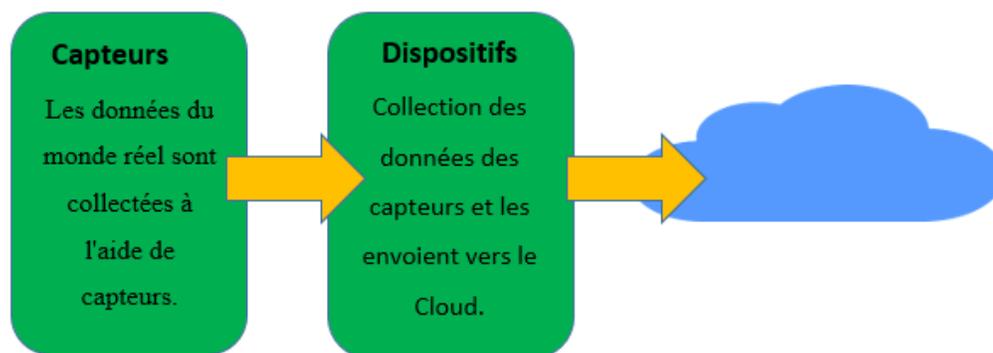


FIGURE II.3 – Communication Capteur/Cloud[1]

II.9.3.1 Généralités sur les capteurs

La réalisation d'un système de robot intelligent mobile est directement liée à la capacité d'apprendre ou de s'adapter à son environnement, les capteurs jouent un grand rôle dans ce domaine à cause de sa capacité de perception requise pour prendre

des décisions intelligentes. L'importance des capteurs dans un travail robotique réside dans le contrôle strict d'un robot en mouvement, connaissant à la fois l'état de l'environnement et l'état du système lui-même, ce que permet un bon suivi du parcours prédéterminé et un meilleur acheminement du travail assigné, correspondant au terrain d'intérêt pour le phénomène capté[39].

II.9.3.1.1 Définition d'un capteur Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en captant un phénomène et de le restituer en un signal c'est à dire une grandeur utilisable, l'objectif d'un capteur et d'un appareil IoT est une longue durée de vie avec peu d'interactions humaines.

Exemple : Un capteur solaire convertissant l'énergie solaire en énergie thermique ou électrique.

II.9.3.1.2 Réseaux de capteurs sans fils (En anglais : Wireless Sensors Network (WSN)), un RSCF c'est un réseau ad hoc qui se compose d'un ou plusieurs nœuds capteurs qui sont organisés en champs « Sensor Fields ». Chacun de ces nœuds a la capacité de collecter des données et de les transférer au nœud passerelle ou puits (dit "sink" en anglais) par l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite ces données par internet ou par satellite à l'ordinateur central « Gestionnaire des tâches » pour analyser ces données et prendre des décisions.

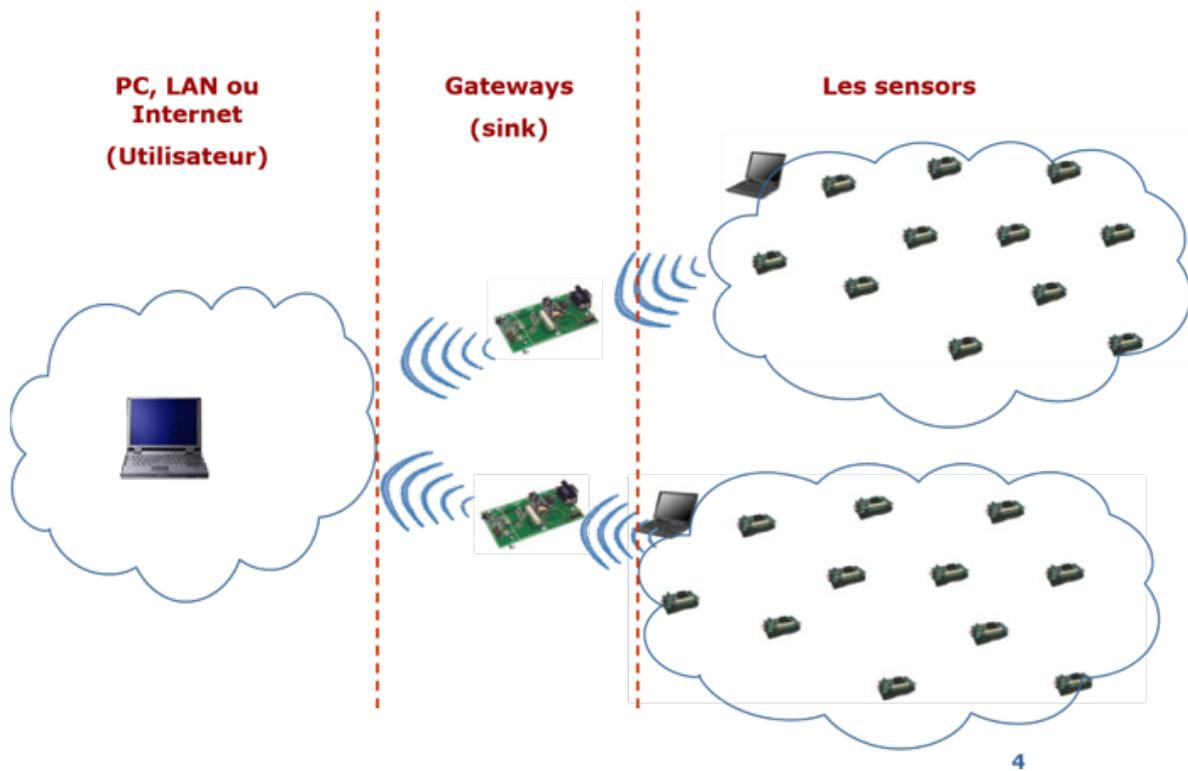


FIGURE II.4 – Architecture du WSN[9].

II.9.3.1.3 Caractéristiques des réseaux de capteurs

- La puissance de traitement et de stockage.
- La qualité de service (QoS).
- La capacité et le type de communication.
- La collaboration entre les nœuds.
- La scalabilité.
- Le déploiement.
- La tolérance aux pannes[23].

II.9.3.1.4 Types des nœuds capteurs

- Noeud régulier : il se compose d'une unité de traitement de données et une unité de transmission qui est responsable de toutes émissions et réceptions de données via un support sans fil.
- Noeud capteur : est un nœud régulier équipé d'une unité de détection qui doté d'un capteur ou plusieurs capteurs pour relever des mesures analogiques et un convertisseur Analogique/Numérique pour rendre l'information relevée en un signal numérique compréhensible par l'unité de traitement.
- Noeud actionneur ou robot : est un nœud régulier avec une unité d'exécution de certaines tâches difficiles (lutte contre les incendies, déplacement, manipulation des produits, etc.).
- Noeud puits : Les puits ont des fonctionnalités plus avancées que les nœuds de détection en termes de transmissions de données et les capacités de traitement, réserves d'énergie avec une mémoire puissante[14].

II.9.3.1.5 Domaines d'applications des réseaux de capteurs sans fils Un RCSF peuvent être appliqués dans différents domaines tels que applications civiles, il existe aussi des applications militaires aux réseaux de capteurs (détection d'intrusions, localisation de combattants, véhicules, armes, etc.), en outre des applications commerciales pour améliorer la procédure de livraison et de stockage, etc.

- **Applications environnementales** : c'est un grand champ, il englobe la surveillance de l'état de l'air ambiant, la qualité des eaux (par exemple le contrôle de leur degré de pollution), le contrôle des environnements dangereux surtout ceux qui sont vulnérables aux incendies, inondation ou glissement de terrain.
- **Applications militaires** : la surveillance de tous les mouvements (amis ou ennemis), analyser le terrain avant d'y envoyer des troupes (détection d'agents

chimiques, biologiques ou de radiations).

- **Applications commerciales** : l'objectif d'utiliser des capteurs dans ce domaine est l'amélioration de processus de stockage et de livraison, par exemple un client passe sa commande et il s'attend jusqu'à la livraison, les réseaux de capteur suivre le procédé de production à partir des matières premières jusqu'au produit final livré.
- **Applications médicales** : il existe des capteurs qui fonctionnent dans le corps humain pour traiter certaines maladies, par exemple un projet de création d'une rétine artificielle composée de 100 micro-capteurs pour correction de vision. D'autres applications biomédicales sont également présentées, telles que : la surveillance de la glycémie, la surveillance des organes vitaux ou la détection précoce des cancers[14].

II.9.3.1.6 Classifications des capteurs Les capteurs sont classés selon les phénomènes qu'ils mesurent : force, énergie, distance, champ magnétique. Il existe de nombreux capteurs disponibles pour l'IoT et plusieurs façons de les catégoriser et les classer[27].

II.9.3.1.6.1 Types de capteur Les capteurs peuvent être divisés en fonction de leurs besoins en alimentation externe[35].

TABLE II.1 – Types des capteurs

Type	Définition	Exemple
Passive	Ne nécessite pas d'alimentation externe pour fonctionner. Ils réagissent aux apports de leur environnement.	Un capteur de température qui change de résistance en réponse aux changements de température.
Active	Nécessite une alimentation externe pour fonctionner.	Une caméra.

II.9.3.1.6.2 Type de signal produit par le capteur Les capteurs peuvent être classés par type de signal, qu'il soit analogique ou numérique [15] :

TABLE II.2 – Types de signal

Type	Définition	Exemple
Analogique	Émettre un signal continu analogique.	Accéléromètres, capteurs de température.
Numérique	La sortie est convertie en valeurs discrètes (1 et 0 numériques) avant d'être transmise à un appareil.	Capteurs numériques de : températures et de pression.

II.9.3.1.6.3 Type d'appareil de mesure Les capteurs peuvent être classés par type d'appareil de mesure, qu'il soit chimique, mécanique ou électrique [22] :

TABLE II.3 – Type d'appareil de mesure

Type	Définition	Exemple
Chimique	Il répond aux changements chimiques de son environnement.	Capteur de gaz.
Mécanique	Il répond aux changements physiques de son environnement.	Micro-interrupteur.
Electrique	Il répond aux changements électriques de son environnement.	Capteur optique.

II.9.3.2 Choix de capteurs

Lors de la sélection d'un capteur IoT, il y a plusieurs choses à considérer. En règle générale. Vous prévoyez de placer des capteurs et d'appareils IoT dans l'environnement souhaité et de les faire fonctionner pendant une période prolongée. Ils peuvent se trouver dans un endroit éloigné ou être intégrés profondément dans un système, inaccessibles aux humains. Le remplacement d'un capteur et d'un appareil dans cette situation peut être extrêmement coûteux, dangereux, voire impossible; toutes les raisons d'examiner attentivement vos décisions en matière de capteurs et d'appareils. La décision est basée sur de nombreux facteurs. Lorsque vous concevez votre système, vous devez examiner attentivement l'importance de chaque facteur et sa priorité par rapport à la conception globale. La liste de considérations suivantes peut être considérée comme un point de départ pour toute discussion sur les capteurs IoT[27].

II.9.3.2.1 Durabilité La durabilité doit être considérée par rapport à l'environnement du capteur. il faut assurer que l'appareil est aussi durable que nécessaire

pour fonctionner pendant une période raisonnable, sans encourir de coûts inutiles. Par exemple, un capteur de température résistant à l'eau peut être acceptable pour une station météo à distance, mais il serait totalement inadapté pour surveiller la température de l'eau dans une piscine car il n'est pas étanche[17].

II.9.3.2.2 Précision Vous voulez avoir suffisamment de précision pour surveiller correctement un environnement, mais vous voulez payer que ce que vous avez besoin. Par exemple, si vous concevez un système pour réguler la température dans une unité de stockage domestique à distance, vous êtes probablement prêt à accepter un capteur qui pourrait être précis à +/- 2 degrés. Cette précision serait totalement inacceptable si vous conceviez un système de dispositif médical. Un capteur de température pour dispositif médical devrait être précis à +/- 0,2 degré[2].

II.9.3.2.3 Polyvalence Les capteurs doivent pouvoir fonctionner dans des variations raisonnables de l'environnement. La plupart des conceptions de réseaux IoT comportent de nombreux capteurs, dans différents d'environnements, il est important d'avoir des capteurs qui peuvent fonctionner avec précision dans toutes les variations de l'environnement[27].

II.9.3.2.4 Consommation d'énergie Selon la situation, vos besoins peuvent concerner un appareil à faible consommation, voire très faible consommation. Vous devrez décider si des fonctions d'économie d'énergie (comme le mode veille ou le réveil rapide) sont nécessaires. Par exemple, un capteur ou un appareil alimenté par des batteries solaires peut devoir passer une grande partie de sa vie en mode veille afin de prolonger la durée de vie de la batterie pendant les périodes de faible luminosité. De plus le temps de réveil rapides pour capturer avec précision les données

est nécessaire[25].

II.9.3.2.5 Considérations environnementales particulières Le choix du capteur peut même affecter la conception finale du système. Par exemple, lors de la conception d'un système de surveillance de la qualité de l'eau, un capteur qui peut être placé dans la canalisation principale d'alimentation en eau est beaucoup plus rentable et précis qu'un capteur qui nécessite de détourner des échantillons d'eau[27].

II.9.3.2.6 Coût Les réseaux IoT impliquent généralement des centaines, voire des milliers de capteurs et d'appareils. Chaque aspect de la conception du capteur doit être considéré en termes de coût. Ces coûts impliquent plus que le prix du capteur. Il faut tenir compte du coût du placement, de la maintenance, de la fiabilité.[29].

II.9.3.3 Inconvénients et contraintes des WSN

- **La sécurité**
 - Établissement et distribution des clés.
 - Authentification des nœuds.
 - Secret.
- **Les contraintes de conception**
 - Plus petit.
 - Moins cher.
- **La consommation d'énergie**
 - Réduction de consommation d'énergie est difficile (calcul, stockage, communication).
- **Durée de vie des capteurs** La durée de vie dépend de l'environnement :
 - Plage de température.

- Vitesse de variation de la température.
- Taux d’humidité.
- Chocs vibration.
- Il existe des méthodes de calcul (ex : MIL HBDK 217F), ça prend en compte aussi la nature du support (circuit imprimé) la technologie de soudure, la nature des vernis de protections[14].

II.9.4 Définition d’un dispositif

Un appareil IoT (dispositifs, devices) peut changer en fonction des besoins du projet. Vous voulez penser aux niveaux d’abstraction lorsque vous concevez votre projet. On peut considérer chaque appareil comme une entité distincte, et d’autres fois où vous souhaitez considérer un groupe de capteurs comme un seul appareil de rapport.

Les exigences spécifiques de votre application vous aideront à comprendre si quelque chose qui génère des informations doit être traité comme un appareil, et mérite donc son propre ID, ou est simplement un canal ou un détail d’état d’un autre appareil [42].

Par exemple, considérons un projet de surveillance de la température dans les chambres d’hôtel. Chaque chambre dispose de trois capteurs : un près du sol, un près du lit et un près du plafond[27].

II.9.4.1 Types d’informations

Chaque appareil peut fournir ou consommer différents types d’informations[1].

II.9.4.1.1 Télémétrie C'est une grandeur physique ou données collectées par l'appareil sont appelées télémétrie. Il s'agit des données visuelles et auditives que les appareils IoT fournissent aux applications. La télémétrie est constituée de données en lecture seule sur l'environnement, généralement collectées via des capteurs.

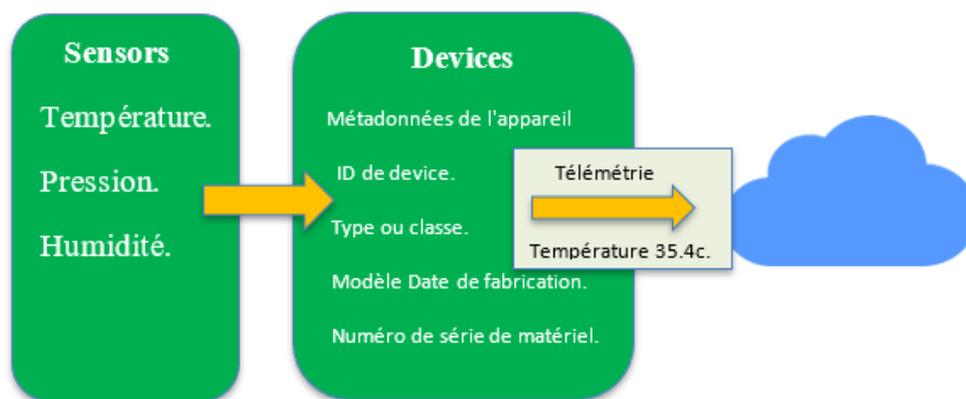


FIGURE II.5 – Type des données (Télémétries) produisent par IoT [1]

II.9.4.1.2 Métadonnées de l'appareil Les métadonnées contiennent des informations sur un appareil. La plupart des métadonnées changent rarement, voire jamais. Une échantillon de champs de métadonnées :

- Identifiant (ID) : Un identifiant qui identifie de manière unique un appareil.
- Classe ou type.
- Modèle.
- Révision.
- Date de fabrication.
- Numéro de série du matériel.

II.9.4.1.3 Commandes de périphérique Les commandes sont des actions effectuées par un appareil. Les commandes peuvent être valides pendant une période

limitée, elles doivent donc inclure une durée de vie (TTL) ou une autre valeur d'expiration. Un ensemble de commandes :

- Tourner à 360 degrés vers la droite.
- Exécutez un cycle d'autonettoyage.
- Augmentez le taux de dix pour cent.

II.9.4.1.4 Informations opérationnelles Les informations opérationnelles sont les données les plus pertinentes pour le fonctionnement de l'appareil par opposition à l'application métier. Cela peut inclure des éléments tels que la température de fonctionnement du processeur et l'état de la batterie. Ce type de données peut ne pas avoir de valeur analytique à long terme, mais il a une valeur à court terme pour aider à maintenir l'état de fonctionnement, comme répondre aux pannes et corriger la dégradation des performances du logiciel après les mises à jour. Ils peuvent être envoyés sous forme de télémétrie ou de données d'état [43].

II.9.5 Motivation sur le Cloud dans l'IoT

L'IoT nécessite des machines très puissantes pour collecter et stocker les données à cause de ses grandes tailles, et le Cloud apparaît donc comme une ressource essentielle dans un projet IoT, il permet de gérer des trafics de données, de traiter ces données, mettre à jour les équipements connectés[1].

II.9.5.1 Cloud Computing

C'est un modèle informatique qui consomme des ressources auprès des fournisseurs sur Internet : serveurs, puissance de calcul, streaming, stockage, etc. Toute l'infrastructure technique est fournie par des fournisseurs de Cloud qui fournissent

ensuite leurs services via Internet. Il s'agit donc d'un nouveau modèle économique utilisant la technologie existante. Il s'agit d'une offre commerciale pour un abonnement économique à un service externe[7] [32].

II.9.5.2 Avantages du Cloud

- Gérance des objets connectés quel que soit leur taille, infrastructure quelconque.
- Capacité de stockage et de traitement d'un grand nombre de données.
- La puissance du serveur est trop élevée.
- Manque de maintenance logicielle et matérielle.
- Un service dématérialisé donc il permet une accessibilité maximale.

II.9.5.3 Limites du Cloud

- **La confidentialité des données** : le fournisseur de service doit garantir un accès aux données qu'aux personnes autorisées.
- **La sécurité des données** : c'est comment je dois récupérer mes données en cas d'accidents, les données peuvent être sauvegardées sur un seul disque ou plusieurs unités de stockage.
- **Le coût et la durabilité du Cloud** : la durée de vie de fournisseur n'est pas garantie.

II.10 Protocoles de communication

Lors de la connexion d'appareils à Google Cloud Platform, vous devrez spécifier le protocole de communication que vos appareils utiliseront. Les choix sont MQTT, HTTP ou les deux[1].

II.10.1 Protocole MQTT

MQTT est un protocole IoT standard de l'industrie (Message Queue Telemetry Transport). Il s'agit d'un protocole de messagerie de publication/abonnement[26].

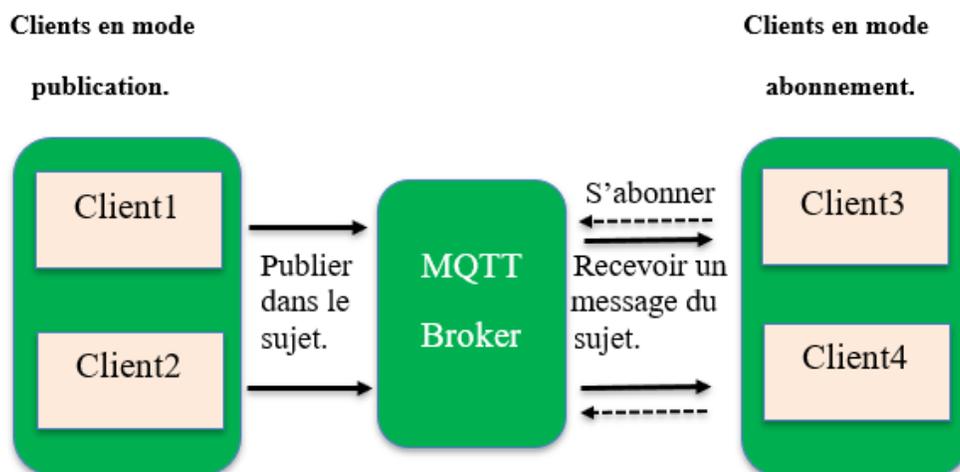


FIGURE II.6 – Architecture du protocole MQTT[1].

II.10.1.1 Principe de fonctionnement du MQTT

MQTT est un modèle client/serveur, où chaque capteur est un client et se connecte à un serveur, appelé courtier (Broker en anglais), via TCP. MQTT est orienté message. Chaque message est un bloc de données discret, opaque pour le courtier. Chaque message est publié à une adresse, appelée sujet. Les clients peuvent s'abonner à plusieurs sujets(Topics). Chaque client abonné à un sujet reçoit chaque message publié dans le sujet(Topic). La taille maximale d'un message envoyé avec MQTT est de 256 Mo. Par exemple, imaginez un réseau simple avec quatre clients et un courtier central. Les quatre clients ouvrent des connexions TCP avec le broker. Les clients 3 et 4 s'abonnent au topic température. Ultérieurement, le client 1 publie une valeur de

22,5 pour la température du sujet. Le courtier transmet le message à tous les clients abonnés[1].

II.10.2 Protocole HTTP

C'est un protocole « sans connexion » : avec le pont HTTP, les appareils ne maintiennent pas de connexion au cloud. Au lieu de cela, ils envoient des demandes et reçoivent des réponses[26].

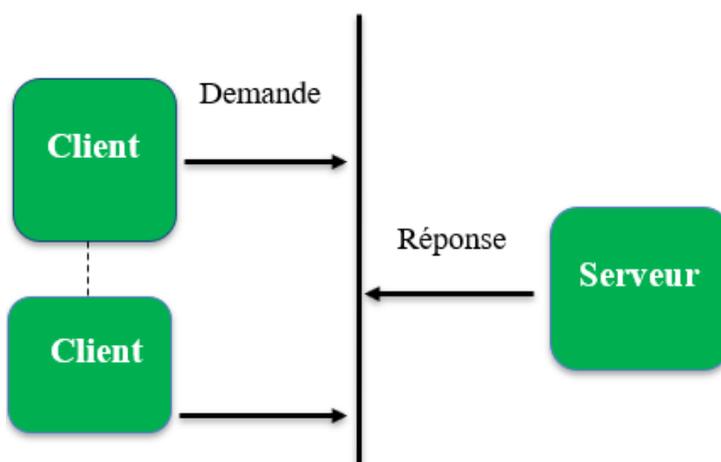


FIGURE II.7 – Architecture du protocole HTTP[1].

II.10.3 Comparaison des fonctionnalités générales des protocoles

Ce tableau résume les explications précédentes sur les deux protocoles [10] :

TABLE II.4 – Différence entre les protocoles MQTT/HTTP

MQTT	HTTP
Transport de télémétrie Message Queuing.	Protocole de transfert hypertexte
Il fonctionne sur le modèle publication/abonnement.	Il fonctionne sur le modèle demande/réponse.
Moins complexe.	Plus complexe.
Il fonctionne sur le protocole de contrôle de transmission.	Il fonctionne sur user datagramme protocole.
La taille du message généré est moindre car il utilise le format binaire.	La taille du message généré est plus élevée car il utilise le format ASCII.
Il fonctionne sur le port de 1883.	Il fonctionne sur le port 80 ou 8080.
Il assure la sécurité des données avec SSL/TLS.	Il n'assure pas la sécurité mais Https est conçu pour cela.
La conception de ce protocole est centrée sur les données.	La conception de ce protocole est centrée sur le document.

II.11 Travaux connexes

Dans cette sous-section, nous mentionnerons certains des travaux antérieurs sur des différents modèles, afin de faire une comparaison entre notre travail et leurs travaux :

Dans [24] , les auteurs propose un modèle mathématique pour étudier et visualiser le mouvement planaire d'un robot quadrupède.

Un modèle cinématique a été créé en écrivant la cinématique avant et arrière du robot. Ils ont été utilisés pour simuler le modèle de marche rampante à l'aide de la

programmation MATLAB et du logiciel de modélisation SolidWorks. Les conceptions de jambes créées dans SolidWorks ont été importées dans MATLAB en tant que fichiers STL (Standard Transformation Language).

Dans [38], et afin d'exploiter les ressources marines et de surveiller les plateformes pétrolières, le concept IdO sous-marins (IoUT) émerge ces dernières années. La plate-forme robotique sous-marine a été largement utilisée dans les systèmes IoUT pour fournir des services, tels que la collecte de données, la livraison, la surveillance et l'exploration sous-marine.

les auteurs ont fait les tâches suivantes :

Katzschmann et al a développé un poisson robotique bionique capable de nager le long d'une trajectoire 3D avec contrôle de flottabilité autonome pour observer aquatique environnements. **Yu et al** conçu le contrôleur hybride pour réaliser un contrôle de profondeur pour un dauphin robotique autopropulsé. **Liao et al** implémenté un poisson robotisé piloté par un fil bionique propulseur battant entraîné. *Morgansen et al* ont développé un robot sous-marin à nage libre capable de produire des mouvements agiles tels que des virages serrés.

II.12 Conclusion

Dans ce chapitre, nous avons couvert certains aspects principaux qui sont à la base de notre travail actuel :IoT, WSN, Cloud, protocoles de communication. Nous avons décrit le rôle et la fonctionnalité de chacun d'eux, et enfin nous avons mentionné certains des travaux connexes précédents. Le chapitre suivant présentera la conception de notre système et la mise en œuvre de ce projet.

Chapitre III

Conception du système

III.1 Introduction

Dans ce chapitre, nous présentons la conception du système, les différents ensembles de données et leurs différentes structures, l'architecture générale et détaillée de notre projet, la phase de prétraitement, nous allons présenter tous ce qui concerne le mécanisme de fonctionnement de robot tels que les équations pour suivre le parcours et chemin de robot. nous expliquerons les algorithmes utilisés et les traduirons en des diagrammes pour décrire le principe de fonctionnement des legs de robot.

III.2 Architecture générale du système

Notre système se compose d'un ensemble de robots qui ont des capteurs pour relever des mesures et des informations pour les transmettre au Dashboard pour faire ce qui est nécessaire.

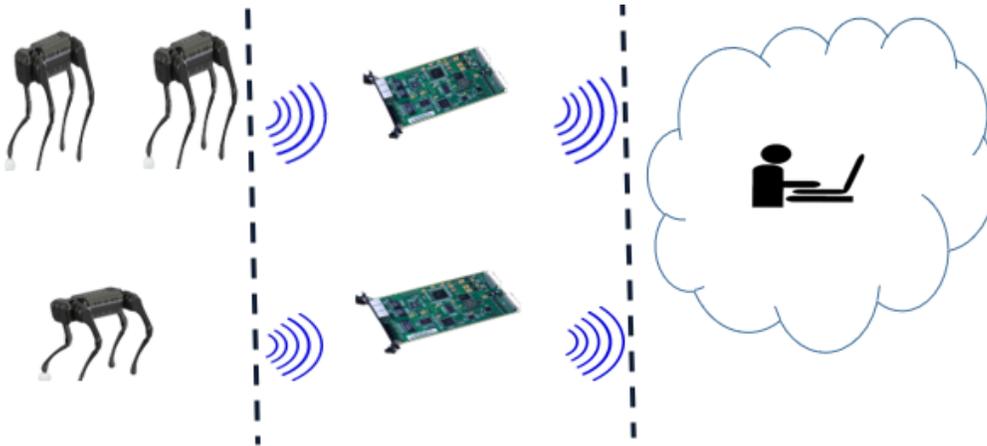


FIGURE III.1 – Architecture générale du système.

III.3 Architecture détaillée du robot

Le robot se compose de cinq modules nécessaires, chaque module a son rôle, dont le premier c'est un module de perception pour la détection dans leur environnement dans lequel il s'interagit, ces informations relevées seront envoyées vers le module de traitement qui inspire leur énergie de panneaux solaires, ce dernier à son rôle va transférer les informations captées vers le module de communication ou de déplacement pour se déplacer.

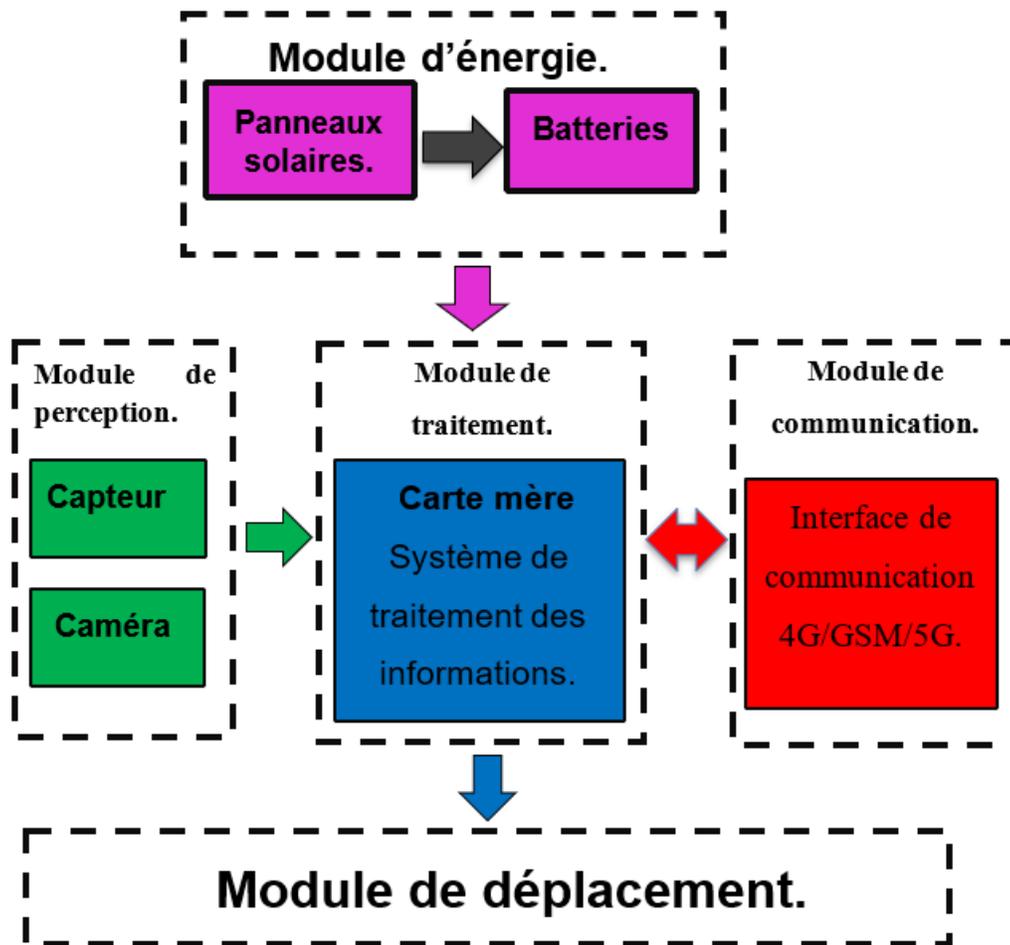


FIGURE III.2 – Arcitecture de robot.

III.3.1 Module de déplacement

Ce module est responsable sur le mouvement et le déplacement de robot, il a un contact direct avec le module du traitement qui le guide après le traitement des informations reçues du module de perception pour le mettre en confortable position en évitant tous les obstacles, on s'intéresse dans ce module au mouvement (les moteurs qui lui permet de bouger) et synchronisations des pieds (legs en anglais).

III.3.1.1 Mécanisme du Leg

Cependant, des modèles mathématiques du robot doivent être développés pour aider à déterminer les paramètres et caractéristiques importants. Un modèle cinématique a été créé en écrivant la cinématique directe et inverse du robot.

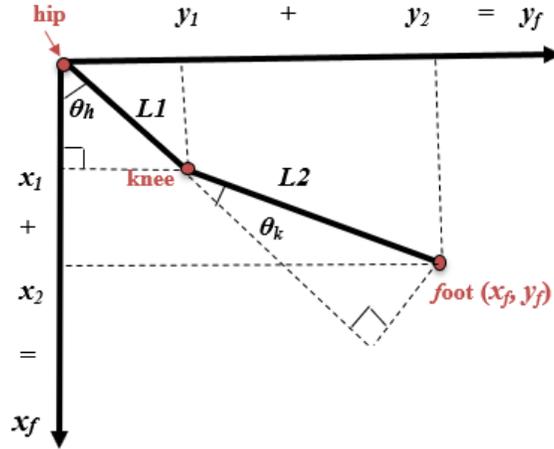


FIGURE III.3 – Modèle de leg implémenté.

III.3.1.1.1 Modèle mathématique de pas En utilisant la cinématique avant pour ce modèle et les paramètres d'articulation connus, Θ_H et Θ_k la position (x_f, y_f) du pied peut être calculée à l'aide de l'équation (1) et de l'équation (2). [24]

$$— x_f = L_1 \times \cos(\Theta_H) + L_2 \times \cos(\Theta_H + \Theta_k) \dots (1)$$

$$— y_f = L_1 \times \sin(\Theta_H) + L_2 \times \sin(\Theta_H + \Theta_k) \dots (2)$$

L'utilisation de la cinématique inverse et d'une position (x_f, y_f) connue de l'effecteur côté pied est connue et les paramètres de l'articulation, Θ_H et Θ_k peuvent être calculés à l'aide de l'équation (3) et de l'équation (4).

$$— \Theta_H = \tan^{-1}\left(\frac{y_k - y_h}{x_k - x_h}\right)$$

$$— \Theta_k = \cos^{-1}\left(\frac{L_1^2 + L_3^2 - L_2^2}{2 \times L_1 \times L_3}\right)$$

III.3.1.2 Algorithmes de déplacement

III.3.1.2.1 Algorithme de leg on générer une procédure pour calculer les points de leg à partir ses paramètres (les longueurs et les angles).

Algorithm 1 procédure $\text{Leg}(\theta_H, \theta_k : \text{Réel})$

```

hip ← point( $X_H, Y_H$ );
knee ← point(hip,  $L_1, \theta_H$ );
newFoot ← point(knee,  $L_2, \theta_H - 180 - \theta_k$ );
if (fixedFoot) then
     $d_x \leftarrow \text{foot}.x - \text{newFoot}.x$ ;
     $d_y \leftarrow \text{foot}.y - \text{newFoot}.y$ ;
    hip.shiftpoint( $d_x, d_y$ );
    knee.shiftpoint( $d_x, d_y$ );
     $X_H \leftarrow X_H + d_x$ ;
     $Y_H \leftarrow Y_H + d_y$ ;
else
    foot ← newFoot;
end if

```

III.3.1.2.2 Algorithme synchronisation des legs Ce algorithme fait la synchronisation entre les quatre leg, il donne l'ordre pour leg_1 et leg_2 pour commencer d'abord avec lecture de fichier Angles qui inculent les valeurs(Θ_H, Θ_k), lorsque la lecture de fichier se termine il va stoper le mouvement des legs leg_1 et leg_4 et il permet au leg_2 et leg_3 de déplacer en retourner Θ_H à leur valeur initiale, et il continue de la même manière les autres pas.

Algorithm 2 Algorithme synchronisationLeg

 $Leg_1 \leftarrow Leg(); Leg_2 \leftarrow Leg(); Leg_3 \leftarrow Leg(); Leg_4 \leftarrow Leg();$ **while** (*true*) **do** $thetaH_1 \leftarrow thetaHList[indice_1]; thetaK_1 \leftarrow thetaKList[indice_1];$ $thetaH_2 \leftarrow thetaHList[indice_2]; thetaK_2 \leftarrow thetaKList[indice_2];$ **if** ($indice_1 < 180$) **then** $Leg_1.fixedFoot \leftarrow false; Leg_4.fixedFoot \leftarrow false;$ **else** $Leg_1.fixedFoot \leftarrow true; Leg_4.fixedFoot \leftarrow true;$ **end if****if** ($indice_2 < 180$) **then** $Leg_2.fixedFoot \leftarrow false; Leg_3.fixedFoot \leftarrow false;$ **else** $Leg_2.fixedFoot \leftarrow true; Leg_3.fixedFoot \leftarrow true;$ **end if** $indice_1 ++; indice_2 ++;$ **end while**

III.3.1.3 Synchronisations entre les legs du robot

Dans ce genre de robot, chaque deux jambes sont actionnées séparément, Les jambes sont actionnées dans l'ordre 1, 4 ensuite 2, 3 en pas transversal Les numéros de pattes sont illustrés à la figure 2.

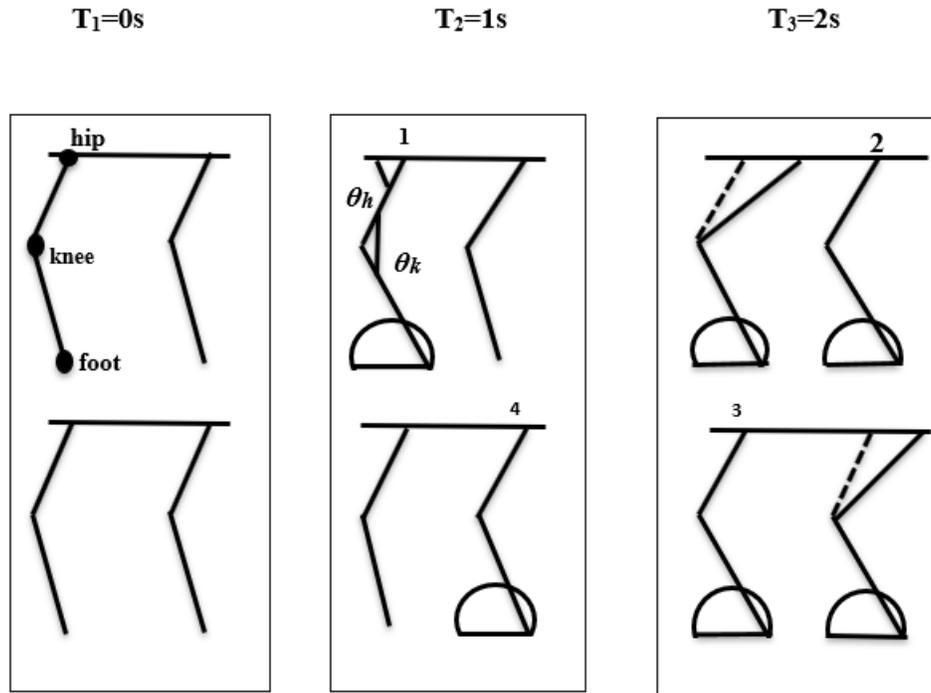


FIGURE III.4 – Synchronisation entre les legs.

III.3.2 Module de perception

Ce module est le responsable pour faire le sensing de l'environnement, dans ce module on utilise un capteur ultrasonique comme des yeux dans le visage de robot, ce capteur calcule la distance entre le robot et un objet pour lui dire s'il peut passer de ce chemin ou non, cette distance relevée va l'envoyer sous forme d'une télémétrie vers le module de traitement (raspberry) pour le traitement.

III.3.3 Module de traitement

Ce module représente l'esprit du robot, il gère tous les autres modules de notre robot, leur objectif est le traitement des télémétries reçues par le module de perception et prendre la décision de la prochaine destination (module de communication ou module de déplacement).

III.3.4 Module de communication

Ce module assure la communication entre le robot et la station de commande à distance (Dashboard), le robot fait le sensing l'environnement et les envoie vers le Module de traitement. Vu que nous vivons dans le désert, nous aurons des problèmes de connectivité, la solution est l'utilisation des cartes sim satellitaires.

III.3.5 Module d'énergie

Dans ce projet on a besoin d'une source d'alimentation électrique pour garantir le fonctionnement du système en permanence et pour cela on a utilisé une énergie solaire (des panneaux solaires qui chargent les batteries).

III.4 Diagramme UML

On a généré les diagrammes à partir du code Java pour bien expliquer et détailler le travail :

III.4.1 Diagramme de fonctionnement

On peut le représenter sous forme de schéma contenant fonction de service, fonctions techniques et des solutions techniques. Après le choix d'un endroit comme un point de départ, on va appliquer trois fonctions et pour chaque fonction on a proposer une solution technique.

Déplacer le robot : par l'alimentation de raspberry.

Diriger le robot : avec deux manières

- **Automatique** : en utilisant un capteur ultrasonique qui calcule la distance et les envoient vers un algorithme A* qui dirige le robot.
- **Manuel** : c'est quelqu'un qui guide le robot à travers une application mobile ou application web.

Acquérir des données : en mesurant quelques paramètres tels que humidité/température en utilisant un capteur DHT 11.

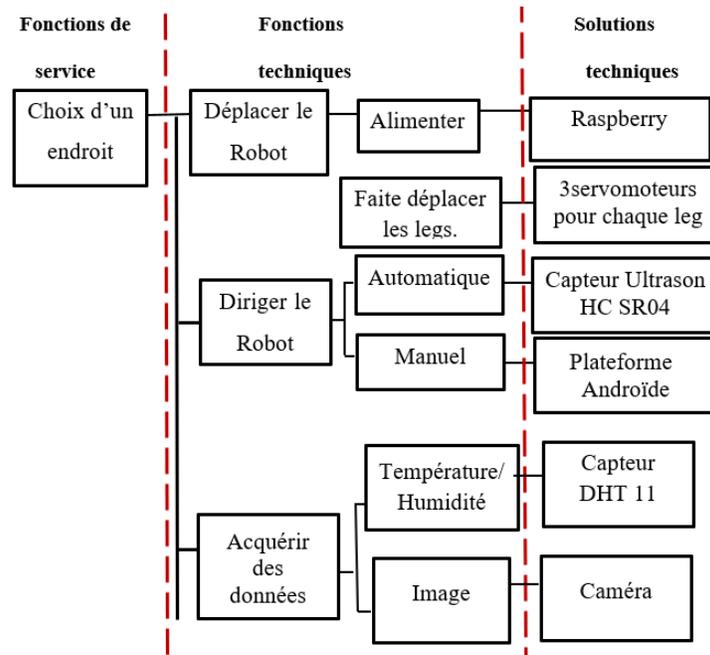


FIGURE III.5 – Diagramme de foctionnement de robot.

III.4.2 Diagramme de classe

On a trois classe dans ce projet Leg,Point et Fennec,une fois on exécute la classe main Fennec, elle fait un appel au classe Leg, la classe leg à son role fait un appel au classe point.

Classe Fennec : contient deux boolean pause pour stoper le mouvement de leg à certains temps, et alive pour lancer et arreter l'exécution,on a crée dans cette classe aussi quatre objets de type leg,les angles et les positions de hip et foot sont déclarés sous forme des listes.

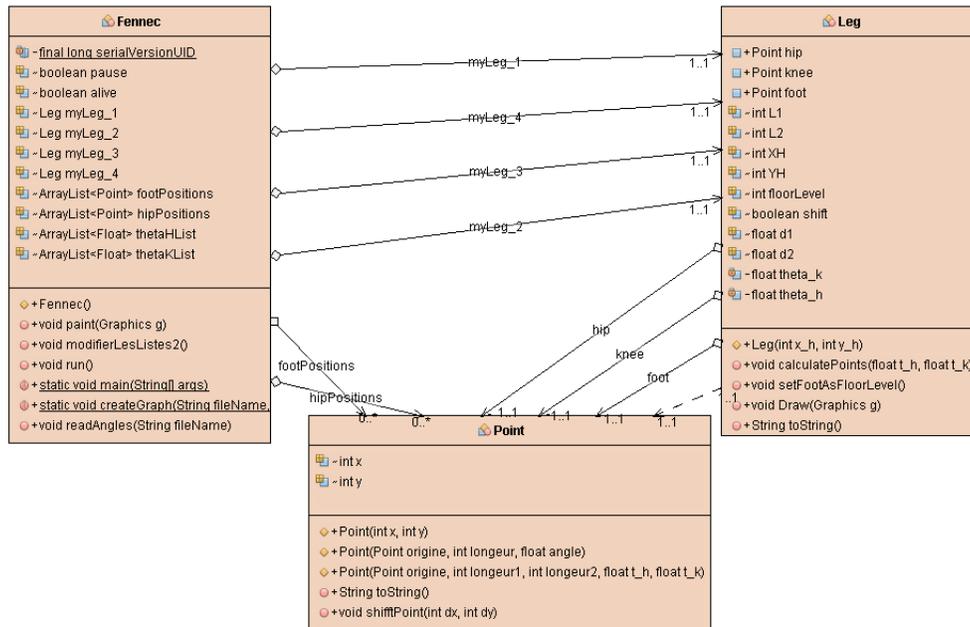


FIGURE III.6 – Diagramme de classe.

Cette classe inclue cinq méthodes :

- Fenec() :contient les quatre legs et la position initial de point hip en entrant n’importe quel valeur.
- paint() :c’est pour tracer legm faire une liason entre hip(hanche), knee(genou), foot(talon).
- readangles() :pour parcourir le fichier généré par le python.
- modifierLeslistes2() :on lit le fichier csv et garde ses valeurs dans des listes pour puisse les modifier selon notre besoin, une fois on finit de parcourir les deux listes de Θ_H, Θ_K on garde la dernière valeur de Θ_K mais Θ_H doit revenir à leur valeur initiale pour que le robot puisse faire un autre pas.
- run() :pour la synchronisation entre les quatre legs,elle fait un appel au fonction modifierliste2(). L’exécution de cette classe commence par la lecture de

fichier angles.csv en tanque alive est vrai, on exécute la méthode `run()` qui fait un appel à la méthode `calculate points` de la classe `leg` puis synchroniser entre les quatre legs.

classe Leg : leurs paramètres sont deux longueurs fixes, un boolean `shift`, trois points `hip`, `newfoot`, `knee`. Cette classe inclut cinq méthodes :

- `leg()` : juste pour des initialisations.
- `calculate points()` : elle contient deux paramètres nécessaires (Θ_H, Θ_k), elle fait un appel à la classe `Point` pour calculer les points.
- `Draw()` : pour déterminer les coordonnées et les couleurs des points.
- `String to string()` : pour convertir les valeurs des deux longueurs et (Θ_H, Θ_k) et les écrire dans un fichier.
- `shiftLeg()` : inclut un boolean `shift` en tanque `foot` se déplace, elle fait une transition linéaire.

Classe Point : contient deux variables `x` et `y` c'est des coordonnées. Elle contient une méthode principale :

- `point()` : on lui donne deux longueurs et les deux valeurs des angles pour calculer le point.

III.4.3 Diagramme de séquence

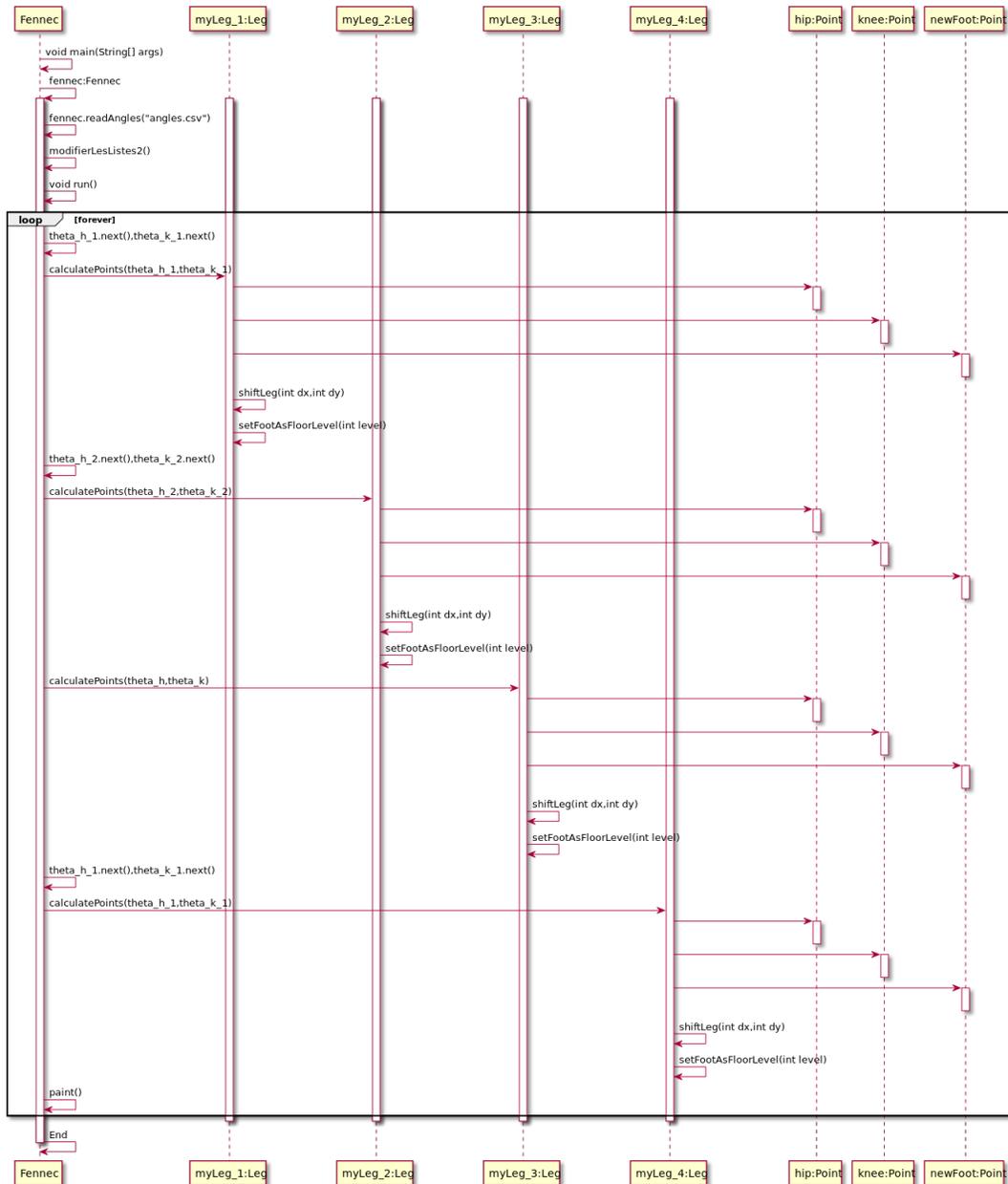


FIGURE III.7 – Diagramme de séquence.

III.5 Conclusion

Dans ce chapitre, nous avons décrit également la conception de notre système, nous avons aussi présenté les différentes méthodes de développement de notre système, ce que nous avons fait exactement dans la phase de prétraitement, et nous avons détaillé notre modèle de robot, leur architecture et structure. Dans le chapitre suivant, nous discuterons des différentes expérimentations et des résultats obtenus.

Chapitre IV

Implémentation du système

IV.1 Introduction

Dans le chapitre précédent, nous avons décrit la conception de notre système. Dans ce chapitre, nous précisons les outils, les bibliothèques et les frameworks utilisés et nous allons présenter le code pour chaque phase de conception du système. À la fin de chaque sous-section, nous discuterons de nos expériences et des résultats obtenus. Enfin, nous comparerons nos résultats obtenus avec certains des travaux connexes précédents.

IV.2 Environnement de développement

Un ensemble d'outils pour aider à améliorer la productivité des programmeurs développant des logiciels :

IV.2.1 Langages de programmation et Framework

Ici, on va spécifier et citer les langages utilisés afin de réaliser ce projet :

IV.2.1.1 Python

C'est langage de programmation orienté objet, open source et populaire à cause de sa simplicité,il est puissant. on peut l'utiliser dans dans nombreux domaines tels que développement des applications,écriture de script[37].

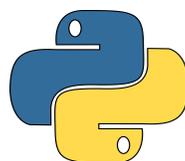


FIGURE IV.1 – Logo python.

IV.2.1.2 Java

C'est langage de programmation orienté objet, open source,il est rapide, sécurisé et fiable, Les applications développées en Java peuvent fonctionner sur différents systèmes d'exploitations, comme Windows ou Mac OS. Avec java on peut simuler et développer des applications mobiles ou bureautique[4].



FIGURE IV.2 – Logo java.

IV.2.2 Outils de développement

IV.2.2.1 Netbeans

NetBeans est un environnement de développement intégré (IDE), placé en open source. NetBeans supporte nativement une variété de langages tels que C, C++, JavaScript, XML, PHP et HTML, ou d'autres. L'interface de netbeans se compose d'un éditeur de code, d'un navigateur de projet, d'un fichier ou d'un service, d'un éditeur de source et enfin d'un éditeur de propriétés CSS. Les utilisateurs n'auront aucun problème à organiser et à ajouter les fenêtres dont ils ont besoin pour l'utiliser[19].



FIGURE IV.3 – Logo Netbeans.

IV.2.2.1.1 JDK Le kit de développement Java fait référence à un ensemble de bibliothèques logicielles de base pour le langage de programmation Java et d'outils capables de compiler du code Java en code d'octet pour la machine virtuelle Java. Il existe plusieurs éditions du JDK, selon la plate-forme Java considérée[21].

IV.2.2.1.2 JVM la JVM est une spécification qui décrit les exigences pour créer un logiciel. Du point de vue des développeurs, la JVM simule l'exécution de programmes Java, c'est un dispositif informatique fictif qui exécute un programme compilé en tant que bytecode Java[13].

IV.2.2.2 Visual Studio Code

VS code est un éditeur de code extensible développé par Microsoft, Il est multiplateforme et supporte des syntaxes pour un grand nombre de langages :HTML, CSS, JavaScript, JSON, Java, SQL, et beaucoup d'autres langages[6].



FIGURE IV.4 – Logo Vs code.

IV.3 Mécanisme du Leg

Pour chaque leg on a trois points(hip, knee,foot) et deux angles(Θ_H, Θ_k). Nous avons simulé un leg à l'aide de python, en donnant les positions pour récupérer le fichier csv des angles qui contient un ensemble de valeurs(Θ_H, Θ_k). L'objectif de python simulation c'est de trouver un intervalle des angles pour que leg puisse faire un pas sans écrasement. La première partie de code c'est pour définir les points et calculé (Θ_H, Θ_k) à partir d'eux.

```
1 def __init__(self, xh, yh, foot: Point) -> None:
2     if foot is None:
3         self.legIsValide = False
4         return
5     self.XH = xh
6     self.YH = yh
7     self.p2 = Point(self.XH + self.X, self.YH, RED)
8     self.p6 = foot
9     self.p6.color = PURPLE
```

```

10     self.legIsValide = self.isTriangle(self.L1, self.L2, Distance(self.p2,
self.p6))
11     if self.legIsValide:
12         self.d1 = round(Distance(self.p2, self.p4), 3)
13         self.d2 = round(Distance(self.p4, self.p6), 3)
14     def calculeAngle(self, length1, length2, length3):
15         numerator = (length1 ** 2) - (length2 ** 2) - (length3 ** 2)
16         denominator = -2 * length2 * length3
17         angle = math.acos(numerator / denominator) # in radians
18         return math.degrees(angle)
19     def calculeAngleDroit(self, pointa: Point, pointb: Point):
20         if(pointa.x == pointb.x):
21             angle = 90
22         else:
23             dx = pointa.x - pointb.x
24             dy = pointa.y - pointb.y
25             pente = -dy / dx
26             angle = math.degrees(math.atan(pente))
27         if(pointa.x > pointb.x):
28             angle += 180
29         return angle
30

```

Listing IV.1 – Simulation leg avec python.

1. Calcule des points

C'est une fonction qui retourne un point, et nous avons utilisé la cinématique directe et inverse que nous avons mentionnée et écrit dans le troisième chapitre.

```

1     def calculePoint(self, point: Point, angle, longueur):
2         angle = math.radians(angle)

```

```

3     x = point.x + (longueur * math.cos(angle))
4     y = point.y - (longueur * math.sin(angle))
5     return Point(x, y, None)
6

```

Listing IV.2 – Simulation leg avec python.

2. Test de validité

Les deux fonctions suivantes c'est pour tester les trois points s'ils forment un triangle ou non.

```

1     def isTriangle(self, length1: float, length2: float, length3:
float):
2         if length1 <= 0 or length2 <= 0 or length3 <= 0:
3             return False
4         if length1 > length2 + length3:
5             return False
6         if length1 < abs(length2 - length3):
7             return False
8         return True
9     def isTriangle2(self, point1: Point, point2: Point, point3: Point):
10        d1 = Distance(point1, point2)
11        d2 = Distance(point1, point3)
12        d3 = Distance(point2, point3)
13        return self.isTriangle(d1, d2, d3)
14

```

Listing IV.3 – Simulation leg avec python.

3. Calcule de distance

C'est une fonction qui permet de calculer la disance entre deux points.

```

1     def Distance(point1: Point, point2: Point):
2     return math.dist((point1.x, point1.y), (point2.x, point2.y))

```

Listing IV.4 – Simulation leg avec python.

4. Calcul les points d'un pas

Pour faire un pas on va faire un demi cercle et calculer les positions par lesquelles passe notre leg.

```

1     def getPoint(centre: Point, rayon: int, angle: int):
2         angle = math.radians(angle)
3         x = centre.x + (rayon * math.cos(angle))
4         y = centre.y - (rayon * math.sin(angle))
5         return Point(x, y, None)
6

```

Listing IV.5 – Simulation leg avec python.

5. Sauvegardement dans CSV

Finalement nous enregistrerons les résultats obtenus dans un fichier Angles.csv.

```

1     def writeValuesToCSV():
2         with open("angles.csv", "w", newline="") as file:
3             writer = csv.writer(file)
4             for line in allValues:
5                 *_ , theta_h, theta_k = line
6                 writer.writerow([theta_h, theta_k])

```

Listing IV.6 – Simulation leg avec python.

Avec Java on fait le contraire, on lit le fichier extrait et on calcule les positions des points que leg traverse.

```
1 public void calculatePoints(float t_h, float t_k) {
2
3     theta_h = t_h;
4     theta_k = t_k;
5
6     hip = new Point(XH, YH);
7     knee = new Point(hip, L1, theta_h);
8     Point newFoot = new Point(knee, L2, theta_h - 180 - theta_k);
9
10    if(shift) {
11        int dx = foot.x - newFoot.x;
12        int dy = foot.y - newFoot.y;
13
14        foot = newFoot;
15        shiftLeg(dx, dy);
16        XH += dx;
17        YH += dy;
18
19    }else {
20        foot = newFoot;
21    }
22
23    if(floorLevel == -1)
24        setFootAsFloorLevel();
25
26    d1 = Distance(hip, knee);
27    d2 = Distance(knee, foot);
28 }
```

Listing IV.7 – Simulation leg avec java.

IV.4 Résultat

Lorsque nous avons obtenu le résultat souhaité en essayant un seul leg, de la meme manière on a crée les trois autres leg et on a commencé synchroniser entre eux.

IV.4.1 Simulation

Comme on a expliqué précédemment que le langage python est utilisé juste pour trouver un fichier angle.csv pour que les longueurs de chaque leg restent fixes.

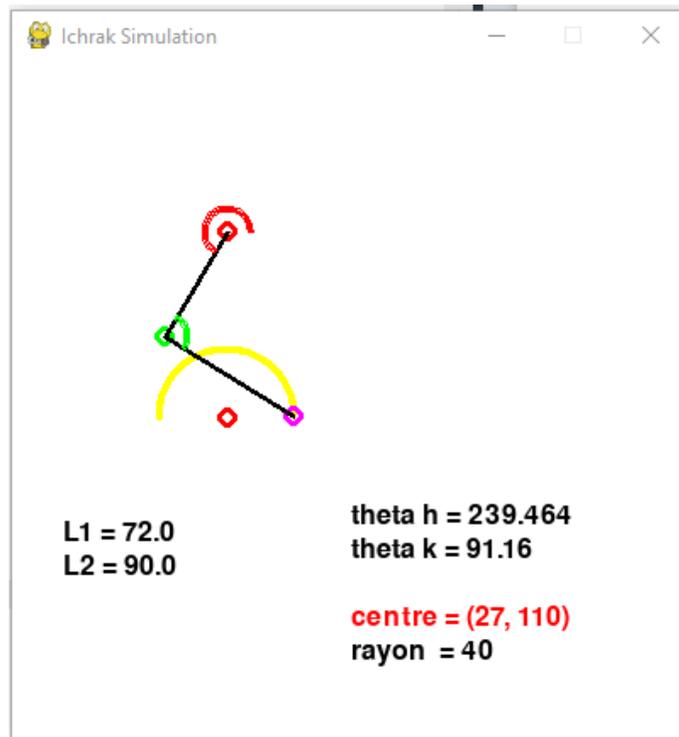


FIGURE IV.5 – Résultat de simulation de leg en python.

1. **Etape 1** :Extraire le fichier Angles.csv Voici le fichier Angles. csv généré par

python

Afin que leg puisse se déplacer on a fait une simulation en python en donnant les positions comme des entrées et on a généré un fichier excel angles.csv en sortie,il contient un intervalle pour les deux angles(Θ_H, Θ_k) afin d'éviter l'écrasement du leg.

	A	B
1	199.794982903925	91.8394397509874
2	199.267926371191	91.1603508592887
3	198.745229459142	90.4814908917785
4	198.226586811085	89.8033710825521
5	197.71068644315	89.1256462134779
6	197.20002540585	88.4484710107336
7	196.693298996379	87.7721037529417
8	196.191449614912	87.0964539568644
9	195.693987091613	86.4223606584877
10	195.20207482547	85.7497257909101
11	194.715048327316	85.077192041177
12	194.233492890484	84.4072052902496
13	193.756194845767	83.7384118481756
14	193.284855728204	83.070732775733
15	192.820032730186	82.4049886082019
16	192.360149794375	81.7410956935406

FIGURE IV.6 – Ensemble des valeurs extraites du fichier csv.

- Etape 2 :** A la fin de simulation on a trcé les courbes de L_1, L_2 en fonction de Θ_H, Θ_K pour montrer qu'on a trouvé les intervalles de fixation des longueurs en fonction des angles, c'est l'objectif principal de la simulation.

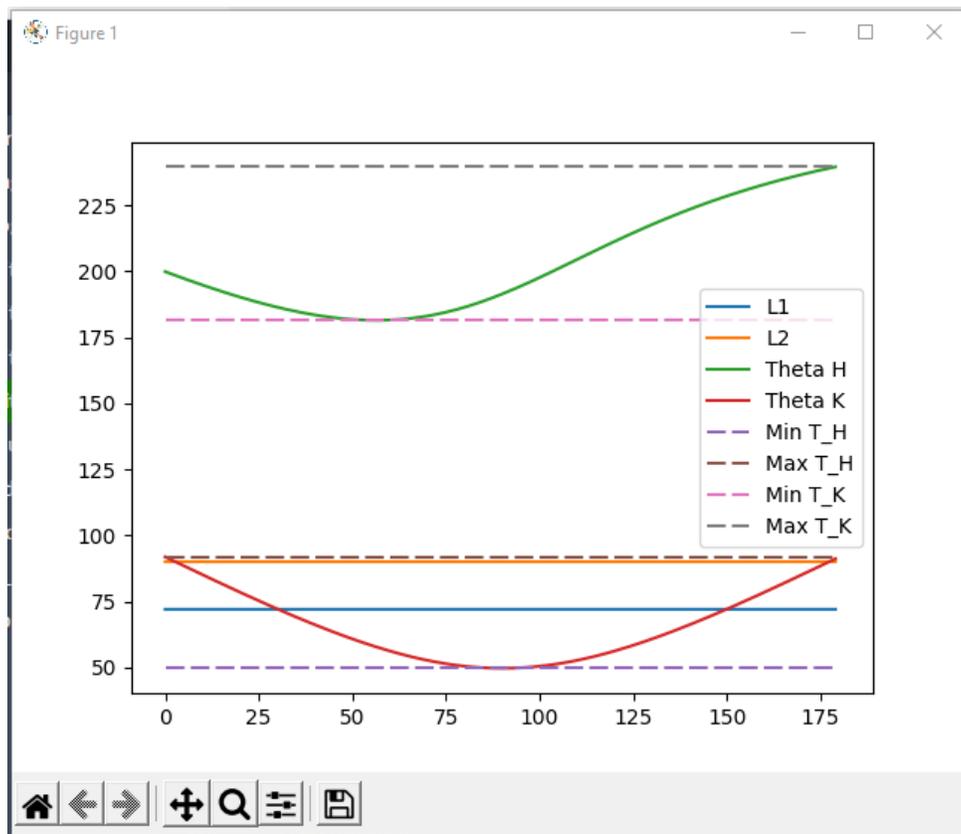


FIGURE IV.7 – Traçage des courbes.

3. **Etape 3** : Ajout de fichier csv dans le code java On a pris le fichier Angles.csv et l'ajouté dans le projet crée en java, et puis on a ajouté une fonction read angles, nous avons aussi ajouté les trois autres leg et synchronisé entre eux, la synchronisation se marche comme suit : on va lire le fichier angles et permettre aux leg_1, leg_4 se déplacent pour faire un pas, les autres deux legs leg_2, leg_3 sont empêchés.

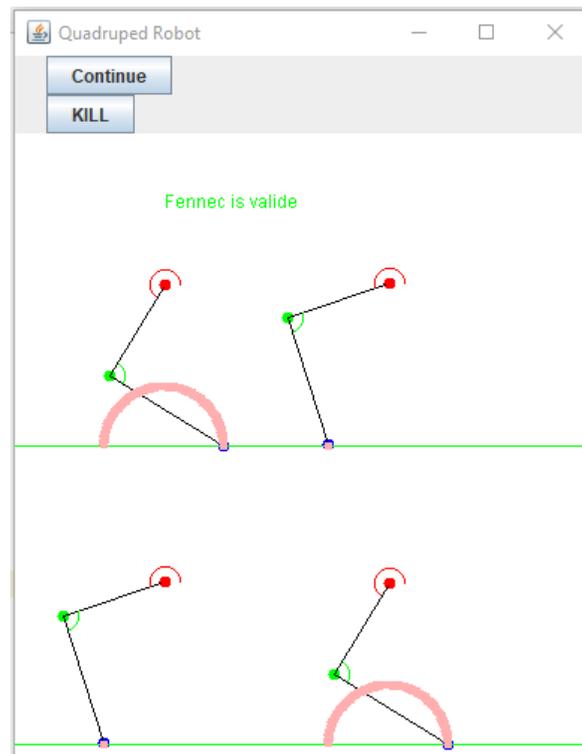


FIGURE IV.8 – Simulation des quatre legs en java.

4. **Etape 4** : Nous avons tracé les courbes pour bien expliquer nos résultats :
 Les résultats du graphe des legs(2,3) montrent que les angles Θ_H , Θ_K sont fixes car les deux legs sont empêchés dans cette étape c'est à dire nous n'avons même pas encore commencé à lire la deuxième liste créée dans le fichier Angles.csv.

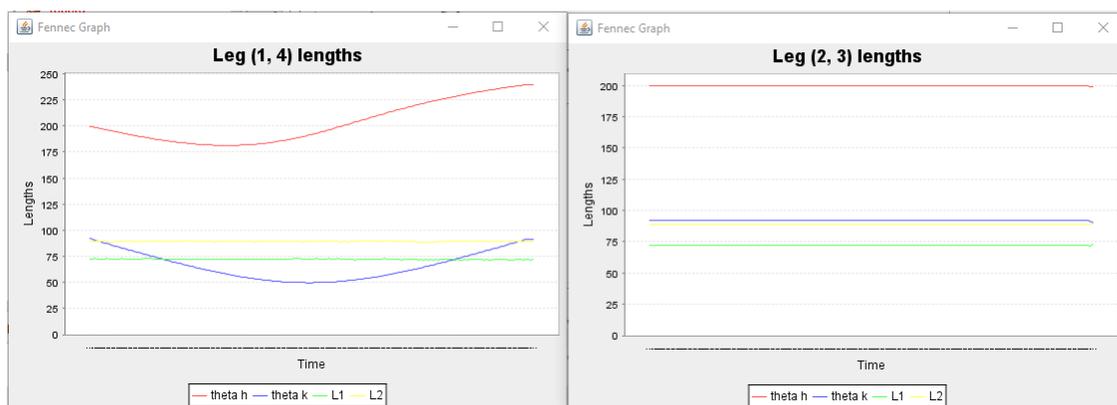


FIGURE IV.9 – Affichage de résultat.

5. **Etape 5** : Une fois leg_1, leg_4 se termine, on va permettre aux leg_2, leg_3 se déplacer en avant pour faire un pas, c'est à dire accomplir un pas avec les quatres legs, durant le déplacement des leg_2, leg_3 on va garder la dernière valeur de Θ_K dans la liste et mais la valeur de Θ_H on doit le remettre à sa valeur initiale.

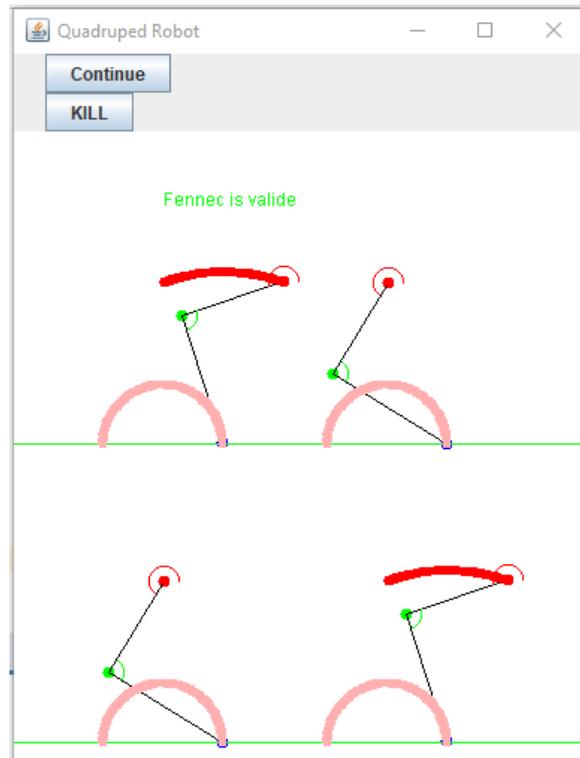


FIGURE IV.10 – Simulation des quatre legs en java.

6. **Etape 6** : Ces courbes affiche ce qu'on a expliqué précédemment : Pour la courbe de leg_1, leg_4 , on remarque que Θ_H a diminué jusqu'à la valeur initiale et la valeur de Θ_K reste fixe.

La courbe de leg_2, leg_3 , on voit que lorsque les deux legs commencent se déplacer les valeurs des angles se change en parcourant la liste de fichier csv.



FIGURE IV.11 – Résultat de simulation.

7. **Etape 7** : L'exécution suivante montrent que les legs leg_1, leg_4 vont faire un deuxième pas, et le point foot de leg_2, leg_3 est empêché au contraire au point hip qui a fait une transition vers l'avant.

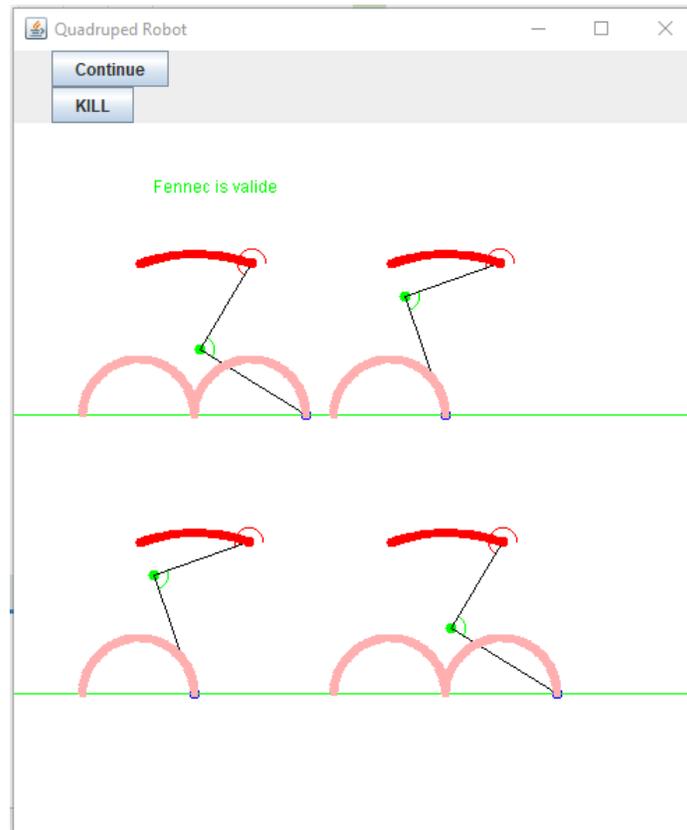


FIGURE IV.12 – Simulation des quatre legs en java.

8. **Etape 8 :** Les courbes de leg_1, leg_4 montrent que on va relire la liste de fichier csv qui contient les valeurs des angles car nous avons remarqué que ces valeurs sont les mêmes qui sont apparues durant le premier pas.

La courbe de leg_2, leg_3 montrent que la valeur de Θ_K va rester fixe jusqu'à la fin de pas des leg_1, leg_4 , et la valeur de Θ_H va diminuer jusqu'à leur valeur initiale.

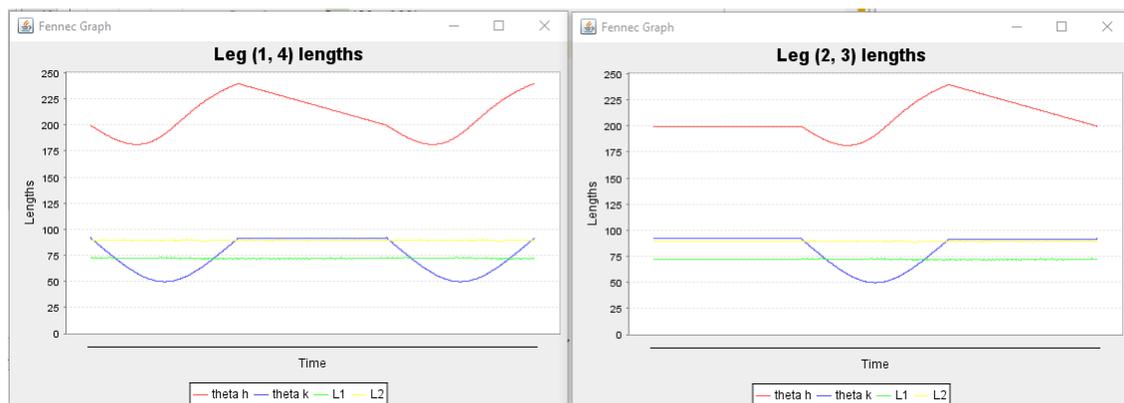


FIGURE IV.13 – Résultat final de simulation d'un pas complet.

IV.4.2 Réalisation

On va présenter notre matériels dans cette section et le résultat final du projet.

IV.4.2.1 Schéma électrique

Dans ce schéma on va présenter le matériel qui sera utilisé dans notre robot.

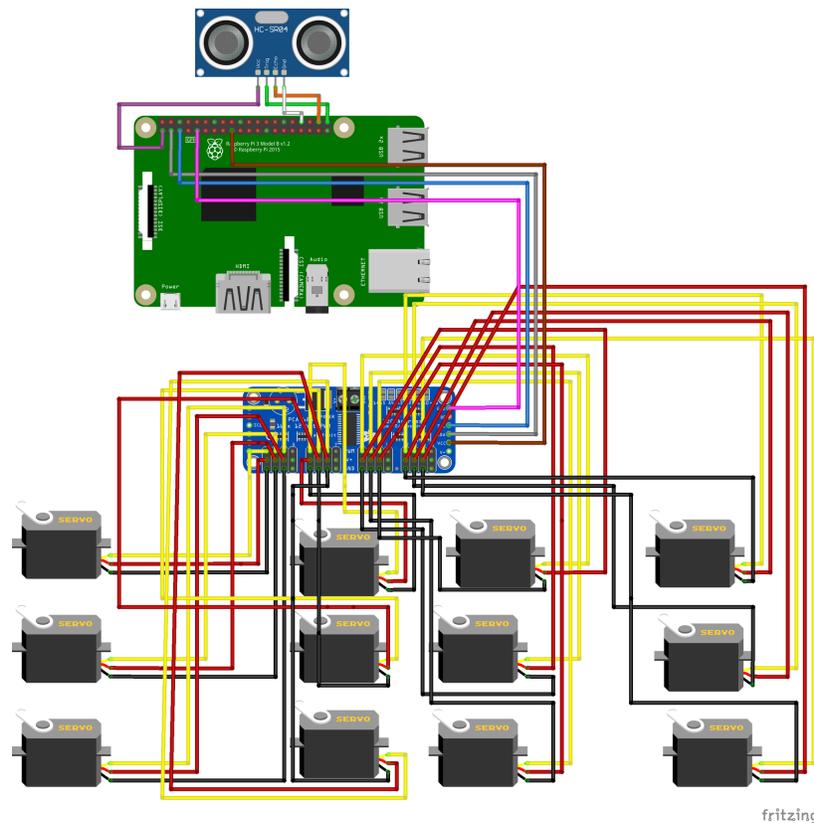


FIGURE IV.14 – Schéma électrique.

IV.4.2.1.1 Servomoteur Les servomoteurs sont des moteurs un peu particuliers, qui peuvent tourner avec une liberté d'environ 180° et garder l'angle de rotation que l'on souhaite obtenir. On utilise des servomoteurs couramment en robotique pour faire des mini robots, ce sont des actionneurs.

Caractéristiques des servomoteurs

1. Dimensions : 32,5 x 12 x 35,5 mm.
2. Tension : de 4.8 à 6 Vcc.
3. Poids : 13,4 g.

4. Vitesse : 0.10 s/60° (4,8 V) / 0.08s/60° (6 V).
5. Couple : 1.8 kg.cm (4,8 V) / 2,2 Kg.cm (6V).
6. Rotation : 180°.
7. Connexion :
 - 3 fils 20 cm : marron (mise à la terre).
 - Rouge (+).
 - Orange (signal).



FIGURE IV.15 – Servomoteur et ses composants.

On a placé trois servomoteurs dans chaque leg de robot, dont le premier contrôle la partie haute du pied (la cuisse), le deuxième pour contrôler la partie basse et le troisième pour gérer le mouvement latérale. Le total des servomoteurs utilisés dans ce robot est douze servos, deux dans chaque pied et pour réaliser ce robot on est besoin de quatre pieds, $4*3=12$.

IV.4.2.1.2 Capteur ultrason C'est un périphérique électronique qui mesure la distance, il utilise l'air comme milieu de propagation. L'émetteur et le récepteur est dans le même boîtier. L'émetteur envoie un train d'ondes Il est reflété dans l'objet découvert (ou travailleur) et renvoyé à la source. Temps requis Pour faire un aller-retour, déterminez la distance de l'objet sauce.



FIGURE IV.16 – Capteur ultrasonique.

IV.4.2.1.3 Raspberry pi Le Raspberry Pi est une taille peu coûteuse, petite et portable de carte informatique. Il peut être utilisé pour se brancher sur un écran d'ordinateur ou télévision, clavier, souris, clé USB, etc. Raspberry Pi a logiciel intégré tel que Scratch qui permet aux utilisateurs de programmer et concevoir une animation, un jeu ou une vidéo intéressante. De plus, les programmeurs peuvent également développer un script ou un programme utiliser le langage Python, c'est le langage principal de Raspbian.

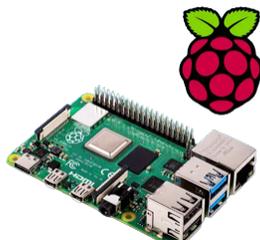


FIGURE IV.17 – Raspberry pi.

IV.4.2.1.4 PCA9685 Les servomoteurs sont souvent entraînés à l'aide des sorties PWM disponibles sur la plupart des microcontrôleurs intégrés. Mais bien que le Pi ait un support matériel natif pour PWM, il n'y a qu'un seul canal PWM disponible pour les utilisateurs sur GPIO18. Ce genre de limite vos options si vous avez besoin de piloter plus d'un servo ou si vous voulez également atténuer une LED ou faire une sorte d'autre bonté PWM également. Heureusement, le PI dispose de HW

I2C, que nous pouvons utiliser pour communiquer avec un pilote PWM comme le PCA9685, utilisé sur le pilote PWM/Servo 12 bits 16 canaux d'Adafruit. En utilisant cette évacion, vous pouvez facilement piloter jusqu'à 16 servomoteurs sur votre Raspberry puis en utilisant notre bibliothèque Python indolore.

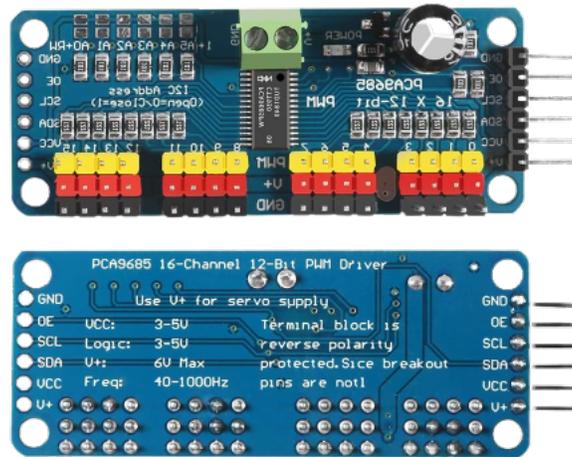


FIGURE IV.18 – PCA9685.

IV.4.2.2 Présentation du matériel dans le robot

Tous les composants qu'on à déjà les présenter précédemment, on va les placer maintenant dans le corps de robot.

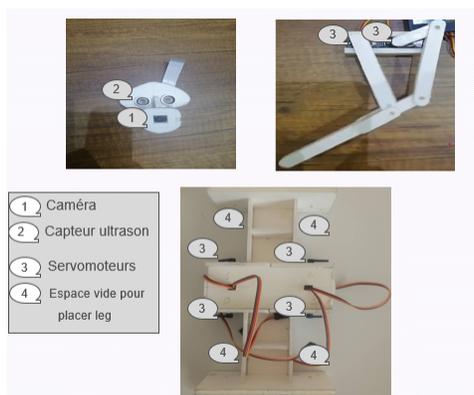


FIGURE IV.19 – Composants de robot.

Voici le résultat de rassembler tous les composants dans un seul corps.

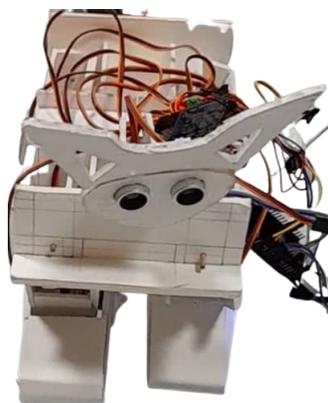


FIGURE IV.20 – Résultat de simulation python.

IV.5 Conclusion

Dans ce chapitre on a deux parties, la première c'est une simulation pour éviter les accidents des matériaux c'est beaucoup plus pour trouver les vrais résultats en essayant virtuellement avant réellement et la deuxième c'est une réalisation en ap-

pliquant les résultats obtenus en les passant dans le matériel. nous avons mentionné les outils, les bibliothèques et les frameworks utilisés,et nous avons montré comment nous avons implémenté notre système.

ous avons aussi expliqué toutes les expérimentations, et on a illustré les résultats par des graphiques.

Conclusion Générale

Nous avons organisé cette mémoire en quatre chapitres, le premier chapitre a contenu des concepts connexes sur les robots, ses caractéristiques, son principe de fonctionnement, ses domaines d'applications, leurs avantages et inconvénients etc. Le deuxième chapitre comprenait des concepts relatifs à une nouvelle technologie (IoT), son fonctionnement, son architecture et ses domaines d'applications etc. Dans le troisième chapitre nous avons présenté l'architecture générale de notre système et détaillée de notre robot et expliqué leur principe de fonctionnement, aussi on a présenté la cinématique de notre modèle géométrique nous avons extrait le diagramme de classe et de séquence à partir de notre code java. Dans le dernier chapitre nous avons présenté tous les résultats obtenus après une simulation avec python puis java.

Les résultats obtenus dans ce projet sont : le déplacement de robot et la synchronisation entre ses legs, la détection et évitement des obstacles.

Concernant les travaux envisagés dans le futur, on peut remplacer le power bank par les panneaux solaires, rendre le robot autonome.

Bibliographie

- [1] Google cloud platform. <https://cloud.google.com/docs>. Online ; accessed 10-February-2022.
- [2] Rachit Agarwal, David Gomez Fernandez, Tarek Elsaleh, Amelie Gyrard, Jorge Lanza, Luis Sanchez, Nikolaos Georgantas, and Valerie Issarny. Unified iot ontology to enable interoperability and federation of testbeds. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 70–75. IEEE, 2016.
- [3] Djebarni Alaeddine. *Etude et conception d'un bras de robot*. PhD thesis, Faculté des Sciences et Technologies, 2021.
- [4] Ken Arnold, James Gosling, and David Holmes. *The Java programming language*. Addison Wesley Professional, 2005.
- [5] Yassine Banouar. *Gestion autonome de la QoS au niveau middleware dans l'IoT*. PhD thesis, Université Paul Sabatier-Toulouse III, 2017.
- [6] Lynda Benyoucef and Nedjma Bouamara. *Conception et réalisation d'une application mobile multiplateforme pour la gestion des ventes*. PhD thesis, Université Mouloud Mammeri, 2020.
- [7] Chamseddine Bouallegue. *DynPubSub : Architecture pair à pair pour les systèmes Pub/Sub orientés-sujets déployés à la périphérie des réseaux*. PhD thesis, École de technologie supérieure, 2021.

- [8] Yacine Challal. *Sécurité de l'Internet des Objets : vers une approche cognitive et systémique*. PhD thesis, Université de Technologie de Compiègne, 2012.
- [9] Yacine Challal, Hatem Bettahar, and Abdelmadjid Bouabdallah. Les réseaux de capteurs (wsn : Wireless sensor networks). *Rapport interne, Université de Technologie de Compiègne, France*, 2008.
- [10] Julien Clément. Objets connectés : stratégies et technologies pour une interaction réussie. 2018.
- [11] Silvio Colombi. Quand la mécatronique réinvente les systèmes mécaniques. Technical report, 1998.
- [12] Rita F De Oliveira, Lysann Damisch, Ernst-Joachim Hossner, Raoul RD Oudejans, Markus Raab, Kirsten G Volz, and A Mark Williams. The bidirectional links between decision making, perception, and action. *Progress in brain research*, 174 :85–93, 2009.
- [13] Nora Douakha and Bouchra Mehadjebia. Une démarche de spécification des besoins et de conception pour la génération automatique d'un code java. 2017.
- [14] Oussama Drissi. *Implémentation d'une stratégie de routage multi-niveau de données d'un réseau de capteurs sans fil dans le domaine ferroviaire*. PhD thesis, Université du Québec à Trois-Rivières, 2014.
- [15] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing : theory and applications*. Cambridge university press, 2012.
- [16] Fatima Zahra Fagroud, Sanaa Elfilali, Hicham Toumi, et al. Iot et cloud computing : état de l'art. In *Colloque sur les Objets et systèmes Connectés*, 2019.
- [17] Rosario Fedele, Massimo Merenda, and Filippo Giammaria. Energy harvesting for iot road monitoring systems. *Instrumentation, Mesure, Metrologie*, 17(4) :605, 2018.

- [18] Mahmoudia Ghilas and Amari Idir. *Commande à distance d'un robot mobile à base d'une Raspberry Pi*. PhD thesis, Université Mouloud Mammeri, 2018.
- [19] Teresa Gomez-Diaz. *Faq : licence & copyright pour les développements de logiciels libres de laboratoires de recherche*, 2009.
- [20] Maaza Hanane. *Conception d'une maison intelligente avec les réseaux M2M/IoT*. PhD thesis, Faculté des Sciences et Technologies, 2021.
- [21] Steve Holzner. *Eclipse*. O'Reilly Media, Inc., 2004.
- [22] Jiří Homola. On the sensitivity of surface plasmon resonance sensors with spectral interrogation. *Sensors and Actuators B : Chemical*, 41(1-3) :207–211, 1997.
- [23] Rahim Kacimi. *Techniques de conservation d'énergie pour les réseaux de capteurs sans fil*. PhD thesis, 2009.
- [24] Kevin Lee. The mathematical model and computer simulation of a quadruped robot. *Research Experience for Undergraduates*, 2014.
- [25] Luca Mainetti, Luigi Patrono, and Antonio Vilei. Evolution of wireless sensor networks towards the internet of things : A survey. In *SoftCOM 2011, 19th international conference on software, telecommunications and computer networks*, pages 1–6. IEEE, 2011.
- [26] AYOUB MARZAK and MOHAMED HAMRAOUI. Architecture de télésurveillance basée sur les réseaux de capteurs sans fils application à la plateforme de capteurs crossbow. *Revue Méditerranéenne des Télécommunications*, 8(2), 2018.
- [27] ZOUAI Meftah. *Une approche cloud computing basée IoT pour le smart House*. PhD thesis, Université de mohamed kheider biskra, 2021.
- [28] Mourad Messaoudi, Takieddine Goual, Kheireddine Lamamra, and Halim Merabti. *Commande intelligente d'une base mobile 4× 4*. 2021.

- [29] Vivek P Mhatre, Catherine Rosenberg, Daniel Kofman, Ravi Mazumdar, and Ness Shroff. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on mobile computing*, 4(1) :4–15, 2005.
- [30] Abdelfetah Mihoubi. Conception et modélisation d’un robot marcheur quadrupède. 2018.
- [31] Sudip Misra, Chandana Roy, and Anandarup Mukherjee. *Introduction to Industrial Internet of Things and Industry 4.0*. CRC Press, 2021.
- [32] Guadalupe Ortiz, Meftah Zouai, Okba Kazar, Alfonso Garcia-de Prado, and Juan Boubeta-Puig. Atmosphere : Context and situational-aware collaborative iot architecture for edge-fog-cloud computing. *Computer Standards & Interfaces*, 79 :103550, 2022.
- [33] Rabeab Saad. *Modèle collaboratif pour l’Internet of Things (IoT)*. PhD thesis, Université du Québec à Chicoutimi, 2016.
- [34] Noureddine Slimane. *Système de localisation pour robots mobiles*. PhD thesis, Université de Batna 2, 2008.
- [35] Stamatis Stamatiadis, Dimitris Taskos, Eleftheria Tsadila, Calliopi Christofides, Christos Tsadilas, and James S Schepers. Comparison of passive and active canopy sensors for the estimation of vine biomass production. *Precision Agriculture*, 11(3) :306–315, 2010.
- [36] Anne-Laure Thessard. Robotisation du travail et compétition symbolique entre les espèces. *Giornale di Filosofia*, (2), 2021.
- [37] Larry Wall, Tom Christiansen, and Jon Orwant. *Programmation en PERL*. ” O’Reilly Media, Inc.”, 2001.
- [38] Chengcai Wang, Jun Lu, Xilun Ding, Chunxiao Jiang, Jianying Yang, and Jianhua Shen. Design, modeling, control, and experiments for a fish-robot-based iot

- platform to enable smart ocean. *IEEE Internet of Things Journal*, 8(11) :9317–9329, 2021.
- [39] Yaser Yousef. *Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil*. PhD thesis, Mulhouse, 2010.
- [40] Meftah Zouai, Okba Kazar, Guadalupe Ortiz Bellot, Belgacem Haba, Nadia Kabachi, and M Krishnamurthy. Ambiance intelligence approach using iot and multi-agent system. *International Journal of Distributed Systems and Technologies (IJDST)*, 10(1) :37–55, 2019.
- [41] Meftah Zouai, Okba Kazar, Belgacem Haba, Guadalupe Ortiz, and Nadia Kabachi. New approach using an iot robot to oversight the smart home environment. *ISTEOpenScience–Published by ISTE Ltd. London, UK–openscience. fr*, 2019.
- [42] Meftah Zouai, Okba Kazar, Belgacem Haba, and Hamza Saouli. Smart house simulation based multi-agent system and internet of things. In *2017 International Conference on Mathematics and Information Technology (ICMIT)*, pages 201–203. IEEE, 2017.
- [43] Meftah Zouai, Guadalupe Ortiz, and Okba Kazar. Paving the way to industry 4.0 : an approach based on multi-agents system and complex event processing.