

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Blida 1

N° D'ordre:



Faculté des Sciences
Département d'Informatique

Mémoire Présenté par :

DJABER Salah HENDI Adel

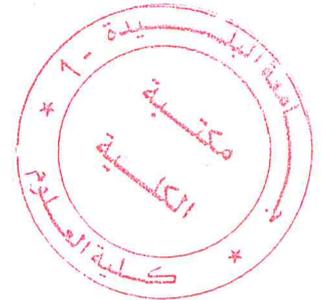
En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : système d'informatique et réseaux



THEME

DEVELOPPEMENT ET IMPLEMENTATION D'UNE COMMANDE A DISTANCE POUR UN ROBOT DE SURVEILLANCE

Organisme d'accueil : Centre de Développement des technologies Avancées (CDTA)

Soutenu le : 27/09/2018

Devant le jury :

M. FERFERA Sofiane	Président
M. HAMMOUDA Mohamed	Examineur
M. CHIKHI Nacim Fateh	Promoteur
M. OUADAH Noureddine	Encadrant
Mme KAHLOUCHE Souhila	Encadrante

Promotion
 2017/ 2018

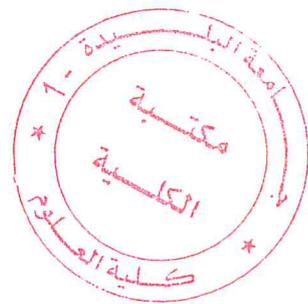
Résumé

La robotique et les systèmes embarqués sont des domaines en pleine évolution que l'on rencontre de plus en plus dans la vie quotidienne.

Dans ce travail, nous nous intéressons au développement et l'implémentation d'une commande à distance pour un robot de surveillance, qui permet aux utilisateurs de capter et contrôler les déplacements du robot dans l'environnement où ce dernier se trouve. Nous avons ainsi développé un système qui contient une application web temps réel qui simplifie l'utilisation à distance d'un robot. Plus précisément, nous avons utilisé Node.js pour la mise en œuvre du serveur, la librairie socket.io pour l'utilisation des websockets, la bibliothèque EasyRTC pour la communication peer-to-peer et le ROS pour la manipulation du robot.

Une telle application permettrait d'éviter à l'utilisateur d'avoir à installer un logiciel ou un plugin sur son ordinateur, tout en lui offrant des fonctionnalités aussi avancées que celles proposées par les logiciels classiques de peer-to-peer.

Mots clés: Application web, Web temps réel, Node.js, ROS, websockets, peer-to-peer, EasyRTC.



Abstract

Robotics and embedded systems are evolving areas and have become ubiquitous in everyday life.

In this work, we have proceeded to the development and the implementation of a remote control for a surveillance robot, which allows users to capture and control the movements of a robot in an environment where he is located.

In this work, we have developed a system that contains a real-time web application that simplifies the remote use of a robot. Specifically, we used Node.js for the server implementation, the socket.io library for websocket usage, the EasyRTC library for peer-to-peer communication, and the ROS for robot manipulation.

Such an application would prevent the user from having to install software on his computer, while offering him features as advanced as those offered by conventional peer-to-peer software. Indeed, unlike the old approaches that required the installation of specific plugins to benefit from a peer-to-peer communication between two browsers.

Keywords: Web Application, Real Time Web, Node.js, ROS, websockets, peer-to-peer, EasyRTC.

إن مجال الروبوتات والأنظمة المدمجة مجال واسع و متطور جدا ، فقد اتخذت مجموعة واسعة من تقنيات التعليم النقطي، فقد أصبحت الروبوتات حاجة أساسية في الحياة اليومية.

في هذا العمل، اخترنا تطبيق مساهمتنا على تطوير وتنفيذ جهاز تحكم عن بعد لروبوت مراقبة، والذي يتيح للمستخدمين الالتقاط والتحكم في تحركات الروبوت في المكان الذي يتواجد فيه، لقد قمنا كذلك بتطوير نظام يحتوي على تطبيق ويب في الوقت الحقيقي، يبسط الاستخدام عن بعد للروبوت.

في هذا المشروع استخدمنا Node.js لتنفيذ الخادم ، ومكتبة socket.io و WebSockets ، ومكتبة EasyRTC للاتصال من نوع نظير إلى نظير (Peer to Peer) و ROS للتحكم بالروبوت.

هذا التطبيق يسمح للمستخدم بتجنب تثبيت برنامج على جهاز الكمبيوتر الخاص به ، مع توفير ميزات له متطورة عن تلك التي تقدمها برامج (peer to peer) التقليدية. في الواقع ، على عكس الطرق القديمة التي تتطلب تركيب المكونات الإضافية المحددة للاستفادة من اتصال نظير إلى نظير (Peer to Peer) بين متصفحين اثنين.

كلمات مفتاحية: تطبيق ويب، ويب في الوقت الحقيقي، EasyRTC, peer-to-peer, WebSockets, ROS , Node.js

Remerciements

En préambule à ce mémoire, je remercie ALLAH qui m'a aidé et donné la patience et le courage durant ces années d'étude. Je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Ces remerciements vont tout d'abord à mon promoteur Mr CHIKHI NACIM pour leur confiance et encouragements.

Je remercie très chaleureusement mes encadreurs Mr. OUADAH NOUREDDINE, et Mme. KAHLOUCHE SOUHILA, pour leurs disponibilités tout en long de la réalisation de ce mémoire, ainsi pour son inspiration, aide et son suivi.

Mes remerciements iront également vers tous ceux qu'ont accepté avec bienveillance de participer au jury de ce mémoire.

Je n'oublie pas mes parents pour leur contribution, soutien et leur patience.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours encouragée au cours de la réalisation de ce travail.

Merci à tous et à toutes

Dédicaces

*Je dédie ce travail à tous les membres de ma très chère
famille, mes amis et mes collègues, en particulier :*

*A Mon cher père qui m'a appris la confiance en soi et
l'autonomie que Dieu le tout puissant te préserve et
t'accorde santé, A ma mère qui était toujours à mes côtés, tu
m'as comblé avec ta tendresse, tes prières ont toujours guidé
mes pas tout au long de mon parcours que le bon Dieu te
donne une longue vie.*

Ma sœur

Mon frère

A tous les membres de la famille DJABAR

A tous mes amis

Et à mon amis Azeddine qui je vien de perdre

Gratitude

SALAH

Dédicaces

Je dédie ce modeste travail

À mes parents qui depuis mon plus jeune âge ont toujours fait leur maximum, en consacrant temps et argent, pour m'éveiller et m'encourager dans mes passions. C'est grâce à vous et pour vous que j'ai fait mon mémoire. Aucun mot sur cette page ne saurait exprimer ce que je vous dois, ni combien je vous aime. Qu'Allah vous bénisse,

vous assiste, vous vienne en aide

À mes chers frères , sœurs et mes neveux

À ma promotrice Mr Chikhi Nassim

À mon encadreur Mr Ouadah Noureddine

À tous mes collègues, plus particulièrement : SALAH,

ABDERRAKIBE, ABDERRAOUF

,ABDELOUAHAB,HOUSSAM,YOUCEF LARBI et tout la section

Master2 SIR en témoignage de mon amitié sincère;

À tous mes amis, plus particulièrement : JALIL, OKACHA,

MUSTAFA, LAMINE,AHMED

en témoignage de mon amitié sincère;

À tous ceux qui m'ont soutenu, qu'ils trouvent ici l'expression de mon

amour et ma profonde

Gratitude

ADEL

Table des Matières

Résumé	i
Abstract	ii
ملخص.....	iii
Remerciements	iv
Dédicaces	v
Table des Matières	vii
Liste des figures	xi
Liste des tableaux	xiii
Liste des abréviations	xiv
Introduction générale	2
Problématique.....	3
Objectif.....	3

Chapiter I: GENERALITE SUR LA ROBOTIQUE

1. Introduction	5
1.1. Définition.....	5
1.2. Les trois lois de la robotique	5
1.3. Le robot	6
2. Généralités sur la robotique mobile	6
2.1. Définition.....	6
2.2. Historique de la robotique.....	6
3. Les composants d'un robot.....	8
3.1. Manipulateur	8
3.2. Effecteur final.....	8
3.3. Actionneurs.....	8
3.4. Capteurs	9
3.5. Contrôleur.....	9
3.6. Processeur	9
3.7. Logiciel	9

4. Les types des robots.....	10
4.1. Les robots manipulateurs	10
▪Les types des robots manipulateurs	11
4.2. Les robots mobiles.....	12
4.2.1.L'architecture des robots mobiles.....	12
4.2.2.Les robots à roues.....	13
5. Domaines d'utilisation des robots	15
5.1. Les robots industriels.....	15
5.2. Robots domestiques ou ménagers	15
5.3. Robots en médecine et chirurgie.....	16
6. Conclusion.....	17

CHAPITRE II: LA COMMANDE A DISTANCE

1. Introduction	19
2. Communications sans fil.....	19
3. Les différents types de communication sans fil.....	20
3.1. Infrarouge	20
3.2. Bluetooth	21
3.3. Wi-Fi.....	23
3.4. ZigBee.....	25
4. Choix de la technologie radio pour la connectivité sans fil des objets utilisés.....	27
4.1. Distance maximale (portée)	27
4.2. Débit :	28
4.3. Autonomie.....	29
5. Conclusion.....	30

CHAPITRE III: CONCEPTION

1. Introduction	32
2. Spécification des besoins.....	32
2.1. Besoins fonctionnels	32
2.1.1.Gestion des utilisateurs:.....	32
2.1.2.Vidéo surveillance:	32

2.1.3. Manipulation du robot:.....	32
2.2. Besoins non fonctionnels.....	33
3. Architecture du système	34
3.1. Communication par vidéo (Media).....	35
3.2. Communication par des messages (Data)	36
4. Modélisation.....	37
4.1. Définition d'UML.....	37
4.2. Les différents types de diagrammes UML	37
4.3. Application du langage de modélisation UML à notre étude.....	38
4.3.1. Définition d'un acteur	38
4.3.2. Les cas d'utilisations:	39
4.3.4. Diagrammes de séquence.....	45
4.3.4. Diagramme de classe	48
5. Conclusion.....	49

CHAPITRE IV : IMPLEMENTATION

1. Introduction	51
2. Outils et langages utilisés.....	52
2.2. JavaScript	52
2.3. Python.....	52
2.4. HTML5.....	52
2.5. CSS	53
3. Implémentation.....	53
3.1. Server NodeJs	53
3.1.1. Node.js	53
3.1.2. Les Modules npm.....	54
3.1.3. Socket.io.....	55
3.1.4. EasyRTC	56
3.2. Le Serveur ROS.....	56
3.2.1. ROS (Robot Operating System).....	56
3.2.2. Transformation des données	57
3.2.3. Web Socket.....	58

4. Présentation de l'application	60
4.1. Interface principale du robot	60
4.2. Interface d'authentification	61
4.3. Interface principale « Accueil »	62
4.4. Interface de Manipulation du robot	63
4.5. Interface « Déconnexion »	64
4.6. Les différents profils (Administrateur/Utilisateur)	64
4.6.1. Menu de l'utilisateur.....	64
4.6.2. Menu de l'administrateur.....	66
5. Conclusion	68
Conclusion générale et perspectives	70
Bibliographie	72

Liste des figures

Figure I- 1: Le robot (ZigBee Controlled PICK & PLACE).....	8
Figure I- 2: les actionneurs.....	8
Figure I- 3: le contrôleur.....	9
Figure I- 4: Structure générale d'un robot industriel.....	11
Figure I- 5: Architecture d'un robot mobile.....	13
Figure I- 6: Robot Unicycle.....	13
Figure I- 7: Robot tricycle.....	14
Figure I- 8: Robot voiture.....	14
Figure I- 9: Robot omnidirectionnel.....	15
Figure II- 1: Le Robot THYMIO II.....	21
Figure II- 2: Le Robot Mbot Blue STEM.....	23
Figure II- 3: Le robot Kompaï.....	24
Figure II- 4: Le robot (ZigBee Controlled PICK & PLACE).....	26
Figure II- 5: Portée comparée (en mètres) des principaux standards de connectivité sans fil.....	27
Figure II- 6: Comparaison de la portée et du débit (en bits/s) des principaux standards de connectivité sans fil.....	28
Figure II- 7: Comparaison de la consommation d'énergie des principaux standards de connectivité sans fil.....	29
Figure III - 1: Architecture générale du système.....	34
Figure III - 2: Architecture de communication par vidéo (Media).....	35
Figure III - 3: Architecture de communication par des messages (Data).....	36
Figure III - 4: Diagramme général des cas d'utilisations.....	39
Figure III - 5: Diagramme de cas d'utilisation « Récupérer le flux vidéo ».....	41
Figure III - 6: Diagramme du cas d'utilisation « Manipuler le robot ».....	42
Figure III - 7: Diagramme du cas d'utilisation « Mettre à jour et maintenance d'application ».....	43
Figure III - 8: Diagramme de séquence « Tourner le robot ».....	45
Figure III - 9: Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur ».....	46
Figure III - 10: Diagramme de séquence du cas d'utilisation « Modifier un utilisateur ».....	47
Figure III - 11: Diagramme de classes.....	48
Figure IV- 1: Structure générale d'un robot.....	51
Figure IV- 2: Liste des Modules utilisés.....	55
Figure IV- 3: Extrait du code d'utilisation de socket.io.....	55
Figure IV- 4: transformation des données entre les nœuds.....	57
Figure IV- 5: Extrait du code d'utilisation de Websocket coté Client.....	58
Figure IV- 6: Extrait du code d'utilisation de Websocket coté serveur ROS.....	59
Figure IV- 7: Interface principale du robot.....	60
Figure IV- 8: Interface « d'authentification ».....	61
Figure IV- 9: Interface « Message d'erreur d'authentification ».....	61
Figure IV- 10: Interface « Principale ».....	62
Figure IV- 11: Interface de Manipulation du robot.....	63
Figure IV- 12: Présentation de barre d'outils.....	63

Figure IV- 13: Interface « Déconnection ».....	64
Figure IV- 14: Interface « Menu de l'utilisateur ».....	64
Figure IV- 15: Interface « Modifier les Information personnel ».....	65
Figure IV- 16: Interface « Modifie le mot de passe ».....	65
Figure IV- 17: Interface « Menu de l'administrateur ».....	66
Figure IV- 18: Interface « Lister les utilisateurs ».....	66
Figure IV- 19: Interface d'inscription d'un nouvel utilisateur.....	67

Liste des tableaux

Tableau III -1: Description textuelle du cas d'utilisation « s'authentifier ».....	40
Tableau III -2: Description textuelle du cas d'utilisation « Récupérer le flux vidéo »	41
Tableau III -3: Description textuelle du cas d'utilisation « manipuler le robot »	42
Tableau III -4: Description textuelle du cas d'utilisation « Modifier les informations ». ...	43
Tableau III -5: Description textuelle du cas d'utilisation « Activer/désactiver les utilisateurs ».....	44
Tableau III -6: Description textuelle de cas d'utilisation « Ajouter utilisateur ».	44

Liste des abréviations

- **IR:** Infrarouge.
- **RF:** Radio Frequency.
- **BLE:** Bluetooth Low Energy.
- **TCP:** Transmission Control Protocol.
- **IP:** Internet Protocol.
- **WPAN:** Wireless Personal Area Network.
- **NFC:** Near Field Communication.
- **CCD:** Charge Coupled Device.
- **IHM:** Interactions Homme-Machines.
- **API:** Application Programming Interface.
- **UML:** Unified Modeling Language.
- **WEB:** World Wide Web.
- **WAE:** WEB Application Extension.
- **IDE:** Integrated Development Environment.
- **UP:** Unified Process.
- **SQL:** Structured Query Language.
- **SGBD:** Système de Gestion de Bases de Données Relationnelles.
- **OMG:** Object Management Group.
- **OMT:** Object Modeling Technique.
- **OOSE:** Object Oriented Software Engineering.
- **MD:** Message Digest.
- **XP:** Extreme Programming.
- **C3:** Chrysler Comprehensive Compensation System.
- **HTTP:** hypertext transfer protocol.
- **HTTPS:** hypertext transfer protocol secure.
- **HTML:** HyperText Markup Language.
- **PHP:** hypertext preprocessor
- **J2EE:** Java 2 Enterprise Edition.
- **LED:** Light-Emitting Diode.
- **ROS:** Robot Operating System.

Introduction générale

Introduction générale

Depuis fort longtemps les êtres humains rêvent de créer des machines intelligentes capables d'effectuer des tâches à leur place. Or créer une machine pouvant réaliser des tâches que seuls les humains sont normalement capables de faire n'est pas simple.

En effet, sans toujours y penser, les tâches les plus élémentaires de la vie quotidienne humaine peuvent devenir extrêmement complexes lorsqu'elles sont analysées de plus près.

La robotique permet d'aider l'homme dans les tâches difficiles, répétitives ou pénibles. De plus elle constitue le rêve de substituer la machine à l'homme dans ces tâches.

Les facultés de perception et de raisonnement des robots progressent chaque jour actuellement et plus encore dans l'avenir, et sont appelés à jouer un rôle de plus en plus important dans notre vie.

La robotique comporte deux grands pôles d'intérêt: la robotique de manipulation (robotique industrielle) et la robotique mobile. Un des problèmes majeurs de la robotique mobile est la planification de mouvement. Autour de ce problème de planification de mouvement de nombreuses études ont été réalisées dans le but de développer des méthodes générales pour guider les robots.

Les robots mobiles comme tout autre catégorie de robots sont appelés à remplacer l'être humain dans des tâches répétitives difficiles ou à risque. Beaucoup de secteurs profitent actuellement des avancées technologiques dans ce domaine.

Dans notre projet nous avons implémenté une application web pour un robot de surveillance permettant à un utilisateur à distance d'interagir en temps réel avec d'autres personnes se trouvant dans un lieu de formation. À partir de son ordinateur, un utilisateur, distant peut à la fois communiquer avec de la vidéo et contrôler les déplacements du robot dans l'environnement où ce dernier se trouve.

Cette technologie offre à l'utilisateur une mobilité, une autonomie et une capacité d'interagir qui résultent en une expérience de téléprésence de grande qualité. Cela accentue l'impression de présence et favorise des échanges gratifiants entre les intervenants impliqués dans les multiples contextes pédagogiques envisageables

Problématique

Le but de cette étude est de trouver un mécanisme permettant à l'utilisateur de se connecter à distance et pouvoir contrôler le robot et récupérer le flux vidéo de la caméra.

Pour cela nous cherchons un moyen rapide et efficace, permettant de contrôler et visualiser directement le chemin de la machine sans aucun problème

Dans le modèle classique de l'application web, le client lance une requête et attend qu'elle soit traitée pour qu'il puisse en faire une autre. On appelle cela le modèle synchrone.

Actuellement et avec le développement des technologies informatiques on a besoin d'une communication entre un PC embarqué et un client. Cela permettrait de faire de la communication audio / vidéo temps réel, en même temps d'envoyer des requêtes au robot pour le manipuler.

Objectif

L'objectif de ce travail est de développer une commande à distance pour un robot de surveillance via internet avec l'établissement d'une communication en temps réel.

Structure du document

Après une introduction générale présentant le contexte de l'étude, la problématique et l'objectif principal, ce document est structuré en quatre grands chapitres :

Chapitre I: Ce chapitre décrit la robotique.

Chapitre II : Ce chapitre décrit la commande à distance des robots.

Chapitre III: Ce chapitre décrit l'analyse des besoins ainsi que la conception de notre système.

Chapitre IV: Ce chapitre décrit la réalisation de notre système.

Ce mémoire est terminé par une conclusion générale et des perspectives.

Chapitre I : Généralités sur la robotique

1. Introduction

La robotique touche aujourd'hui de nombreux secteurs de la vie, elle a extrêmement progressé durant le siècle dernier, nous voyons l'évolution de cette invention au fil du temps.

Avant d'entrer dans le cœur du sujet, il est important d'avoir une idée générale du fonctionnement d'un robot mobile afin de bien comprendre les interactions entre les différents modules.

Dans ce premier chapitre, nous présentons une vue générale sur les robots manipulateurs et les robots mobiles et leurs différents domaines d'utilisation.

1.1. Définition

L'écrivain tchèque, Karel Capek, dans son roman, introduit le mot "robot" au monde en 1921. Il est dérivé du mot tchèque robota qui signifie "travailleur forcé". Isaac Asimov l'écrivain russe de science-fiction, a inventé le mot robotique dans son histoire "Habillage", publié en 1942, pour désigner la science consacrée à l'étude des robots. [1]

Avant de définir qu'est-ce qu'un robot nous citerons les trois lois qui ont été développés par Isaac Asimov, et qui régissent le comportement d'un robot. [2]

1.2. Les trois lois de la robotique

- Un robot ne peut blesser un humain ni, permettre qu'un humain soit blessé.
- Un robot doit obéir aux ordres donnés par les humains, sauf si de tels ordres se trouvent en contradiction avec la première loi.
- Un robot doit protéger sa propre existence aussi longtemps qu'une telle protection n'est pas en contradiction ni avec la première et/ou ni avec la deuxième loi. [2]

1.3.Le robot

C'est une machine pouvant manipuler des objets en réalisant des mouvements variés dictés par un programme aisément modifiable.

Programmer un robot consiste, dans un premier temps, à lui spécifier la séquence des mouvements qu'il devra réaliser.

Certains robots sont dotés de "sens" ; c'est-à-dire d'un ensemble plus ou moins important d'instruments de mesure et d'appréciation (caméra, thermomètre, télémètre, ...) permettant au programme du robot de décider du mouvement le mieux adapté aux conditions extérieures. Par exemple: si un robot mobile muni d'une caméra est amené à se déplacer dans un local inconnu, on peut le programmer pour qu'il contourne tout obstacle qui entraverait sa route.

On essaie également de doter des robots d'un dispositif d'intelligence artificielle afin qu'ils puissent faire face à des situations imprévues et nouvelles (le robot pourrait acquérir une certaine "expérience").

2. Généralités sur la robotique mobile

2.1.Définition

Le terme "Robot" désigne une machine à aspect humain, capable de se mouvoir et d'agir, qu'un mécanisme automatique pouvant effectuer certaines opérations, et capable parfois de modifier de lui-même son cycle de fonctionnement et d'exercer un certain choix. [3]

2.2.Historique de la robotique

Au cours de l'histoire nous pouvons distinguer trois types de robots correspondants en quelque sorte à l'évolution de cette « espèce » créée par l'homme. [3]

Le premier type de machine que l'on peut appeler robot correspond aux « Automates ». Ceux-ci sont généralement programmés à l'avance et permettent d'effectuer des actions répétitives.

Le second type de robot correspond à ceux qui sont équipés de capteurs (en fait les sens du robot). On trouve des capteurs de température, photo électronique, à ultrasons pour par exemple éviter les obstacles et/ou suivre une trajectoire.

Ces capteurs vont permettre au robot une adaptation relative à son environnement afin de prendre en compte des paramètres qui n'auraient pas pu être envisagés lors de leur programmation initiale.

Ces robots sont donc bien plus autonomes que les automates mais nécessitent un investissement en temps de conception et en argent plus conséquent.

Enfin le dernier type de robot existant correspond à ceux disposant d'une intelligence dite « artificielle » et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones.

En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme peut le faire l'être humain. Bien sûr il faudra attendre encore longtemps avant que le plus « intelligent » des robots ne soit égal, tant par sa faculté d'adaptation que par sa prise de décisions, à l'homme.

En 1967, l'institut (Stanford Research Institute) construit le premier robot mobile (Shankey) capable de voir, penser et de réagir au changement dans son environnement.

C'est une machine à roues connectée à un gros ordinateur et qui évolue dans un environnement de cubes et de pyramides de tailles et de couleurs différentes. Ses moyens de perception sont essentiellement une caméra qui lui permet d'acquérir des images de son environnement. Quant à ses performances, ça lui demande une cinquantaine de minutes pour effectuer sa mission. [4]

3. Les composants d'un robot

Un robot, en tant que système, se compose d'éléments, qui sont intégrés ensemble pour former un ensemble. La plupart des robots contiennent les éléments suivants :

3.1. Manipulateur

C'est le corps principal du robot qui comprend les jonctions, les articulations, et d'autres éléments de structure du robot. Il convient de noter ici que le manipulateur seul n'est pas un robot, voir la figure I-1. [5]



Figure I- 1: Le bras manipulateur.

3.2. Effecteur final

Cette partie est reliée à la dernière jonction (main) d'un manipulateur qui gère généralement les objets, établit des connexions à d'autres machines ou effectue les tâches requises. [5]

3.3. Actionneurs

Les actionneurs sont les « muscles » de manipulateurs. Le contrôleur envoie des signaux aux actionneurs, qui, à leur tour, déplacent les articulations du robot et des jonctions, les types communs des actionneurs sont les servomoteurs, les moteurs pas à pas, les actionneurs pneumatiques et les vérins hydrauliques.

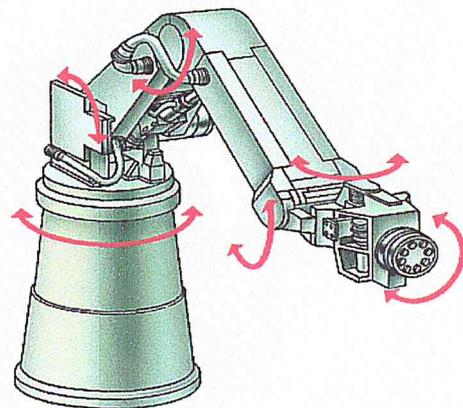


Figure I- 2: les actionneurs

Les actionneurs sont sous le contrôle du contrôleur, voir la figure (Figure I-2). [5]

3.4. Capteurs

Les capteurs sont utilisés pour acquérir des informations sur l'état interne du robot ou pour communiquer avec l'environnement extérieur. Comme chez l'humain, le dispositif de commande de robot doit connaître l'emplacement de chaque lien du robot afin de connaître la configuration du robot. Toujours comme vos principaux sens de la vue, le toucher, l'ouïe, le goût, et la parole, les robots sont équipés de dispositifs sensoriels externes comme un système de vision, le toucher et les capteurs tactiles, synthétiseur de parole, et grâce à eux le robot peut communiquer avec le monde extérieur. [5]

3.5. Contrôleur

Le contrôleur est plutôt proche du cervelet humain ; même s'il n'a pas la puissance du cerveau; il contrôle toujours les mouvements. Le contrôleur reçoit les données de l'ordinateur (le cerveau du système), commande les mouvements des actionneurs, et coordonne les mouvements avec les informations envoyées par les capteurs. [5]

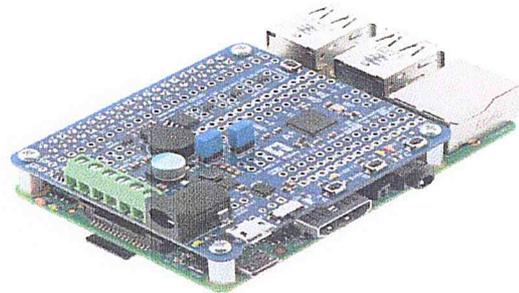


Figure I- 3: le contrôleur.

3.6. Processeur

Le processeur est le cerveau du robot. Il calcule les mouvements des articulations du robot, détermine combien et à quelle vitesse chaque joint doit se déplacer pour atteindre l'emplacement et la vitesse souhaitée, et supervise les actions coordonnées du contrôleur et les capteurs. Dans certains systèmes, le contrôleur et le processeur sont intégrés ensemble en une seule unité, et dans d'autres cas, ce sont des unités séparées. [5]

3.7. Logiciel

Trois groupes de logiciels sont utilisés dans un robot. L'un est le système d'exploitation qui exploite le processeur. Le second est le logiciel robotique qui calcule le mouvement nécessaire de chaque joint du robot basé sur des équations cinématiques. Ces informations sont envoyées au dispositif de commande. Ce logiciel peut être à différents niveaux, de la langue de la machine aux langues sophistiquées utilisées par les robots modernes. Le troisième groupe est la collection d'application - orientée les routines et les programmes développés pour utiliser le robot ou ses périphériques pour des tâches spécifiques telles que l'assemblage, le chargement de machines, la manutention et les routines de vision. [5]

4. Les types des robots

Il existe deux grandes familles de robots qui sont :

- Les robots manipulateurs.
- Les robots mobiles.

4.1. Les robots manipulateurs

Un robot manipulateur est en forme d'un bras et se compose d'un certain nombre de segments qui est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement en toute sécurité. Les robots manipulateurs sont utilisés dans des applications industrielles pour effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage. [6]

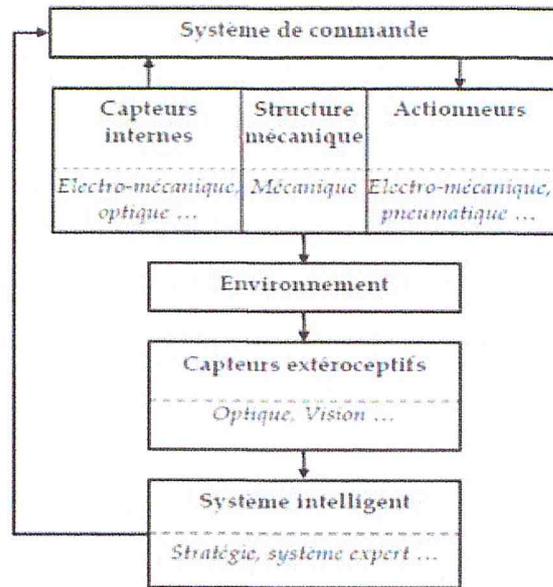


Figure I- 4: Structure générale d'un robot industriel.

▪ **Les types des robots manipulateurs :**

Les robots manipulateurs viennent sous plusieurs formes. Les formes se répartissent en cinq grandes catégories :

- Robots cylindriques
 - Robots rectilignes
 - Robots sphériques
 - Robots articulés
 - Robots SCARA
- a. Robots cylindriques : Le robot cylindrique a deux axes de mouvement, un pour le mouvement en haut et bas. La rotation se fait par la jonction à la base. De plus, le bras horizontal peut se déplacer à l'intérieur et à l'extérieur, ce qui donne un troisième axe de mouvement limitée. [7]
 - b. Robots rectilignes: Les robots rectilignes à trois axes de mouvement (x, y, z). Pour cette raison, le robot rectiligne est parfois appelé Robot cartésien. Ces robots sont exploités par vérin pneumatique. [7]
 - c. Robots sphériques: Le robot sphérique est de grande taille avec un bras télescopique qui assure un mouvement à l'intérieur ou à l'extérieur. Les

L'architecture des robots mobiles est représentée sur la figure suivante :

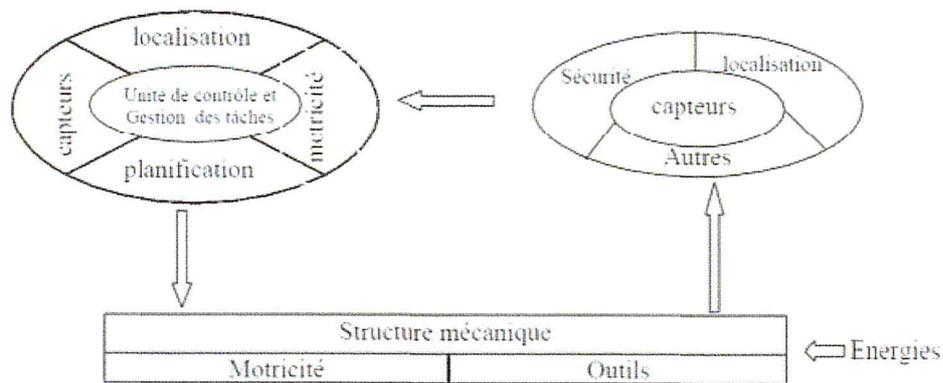


Figure I- 5: Architecture d'un robot mobile.

4.2.2. Les robots à roues

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées.

Nous citerons ici les quatre classes principales de robots à roues.

a. Robot unicycle:

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues spéciales pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

C'est un robot non-holonyme, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites. [11]



Figure I- 6: Robot Unicycle.

b. Robot tricycle:

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable.

C'est un robot non-holonome. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable. [11]



Figure I- 7: Robot tricycle.

c. Robot voiture :

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe.

Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire.

Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé. [11]

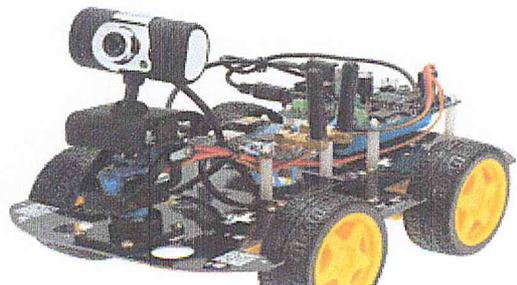


Figure I- 8: Robot voiture.

d. Robot omnidirectionnel :

Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral.

L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande. [11]

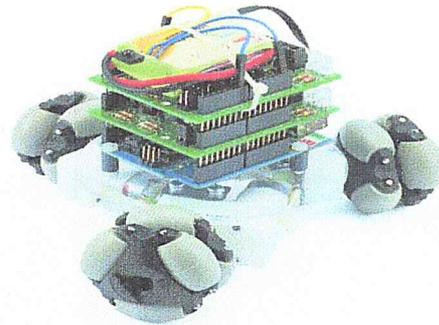


Figure I- 9: Robot omnidirectionnel.

5. Domaines d'utilisation des robots

5.1. Les robots industriels

Ce sont des robots utilisés dans un environnement de fabrication industrielle. Ils sont utilisés dans la fabrication des automobiles, des composants et des pièces électroniques, des médicaments et de nombreux produits

5.2. Robots domestiques ou ménagers

Ils sont utilisés à la maison. Ce type de robots comprend de nombreux appareils très différents, tels que les aspirateurs robotiques, robots nettoyeurs de piscines, balayeuses, nettoyeurs gouttières et autres robots qui peuvent faire différentes tâches. En outre, certains robots de surveillance et de téléprésence pourraient être considérés comme des robots ménagers s'ils sont utilisés dans un tel environnement.

5.3.Robots en médecine et chirurgie

Les robots semblent avoir de l'avenir à l'hôpital. Robodoc aide à réaliser certaines opérations de chirurgie. Le robot infirmier est encore en projet. Le cyber squelette HAL aide les personnes à se déplacer. Et le robot patient permet aux futurs chirurgiens-dentistes d'apprendre à soigner sans faire de dégâts...

- **Le système chirurgical Da Vinci :** Le robot chirurgical permet d'opérer à distance, soit dans la même pièce avec une machine comme intermédiaire, soit d'un endroit très éloigné, ce qui peut être très utile souvent.
- **Les infirmiers du futur :** Les infirmiers qui portent et déplacent les malades seront des robots. En fait, ils ne sont pas prévus pour un avenir si lointain: ils fonctionnent déjà!

Le robot infirmier peut prendre un patient dans ses bras, le porter et le déposer dans un fauteuil.

- **Le robot patient :** L'actroïde Simroid assez réaliste réagit quand l'opérateur le touche à un endroit sensible. Il permet ainsi de réaliser un apprentissage du métier sans frais.

Hanako Showa est une initiative similaire. Grâce à des capteurs implémentés dans ses dents artificielles, le robot peut réagir aux actes du praticien novice, émettre des gémissements ou bouger les bras quand on lui "fait mal". Il peut même communiquer grâce à un procédé de synthèse vocal.

6. Conclusion

A travers ce chapitre on peut dire que dans le cadre de la robotique, la robotique mobile joue un rôle très important. Contrairement aux robots industriels manipulateurs qui travaillent de façon autonome dans un grand nombre d'usines automatisées, les robots mobiles sont très peu répandus. Cette situation n'est pas due au manque d'applications possibles, mais dès qu'on dispose de la mobilité, on peut imaginer des robots facteurs, nettoyeurs, gardiens, démineurs, explorateurs, jardiniers etc. La faible diffusion est surtout due au fait que ces tâches ont une complexité bien supérieure à celles effectuées par des robots manipulateurs industriels. Le monde dans lequel un robot mobile doit se déplacer est souvent très vaste, partiellement ou totalement inconnu, difficilement caractérisable géométriquement et ayant une dynamique propre.

Chapitre II:
La commande à
distance

1. Introduction

Dans cette partie nous présentons la commande à distance d'un robot mobile, en décrivant la communication sans fil qui utilise les ondes (signaux) électromagnétiques pour transmettre des données en utilisant l'air comme canal de transfert. Elle consiste à envoyer ces signaux via une bande de fréquence déterminée, et selon des protocoles déterminés (Wi-Fi, Bluetooth, etc.) .

2. Communications sans fil

Le terme de communication sans fil a été introduit au 19ème siècle et la technologie de communication sans fil s'est développée au cours des années suivantes. C'est l'un des moyens les plus importants de transmission de l'information d'un appareil à d'autres appareils.

De nos jours, le système de communication sans fil est devenu une partie essentielle de divers types de dispositifs de communication sans fil, qui permettent à l'utilisateur de communiquer même depuis des zones opérées à distance.

Dans cette technologie, l'information peut être transmise dans l'air sans nécessiter de câble ou de fils ou d'autres conducteurs électroniques, en utilisant des ondes électromagnétiques comme IR, RF, satellite, etc. À l'heure actuelle, la technologie de communication sans fil fait référence à une variété de dispositifs de communication sans fil et de technologies allant des téléphones intelligents aux ordinateurs, robots, portables, imprimantes. Ce chapitre donne un aperçu de la communication sans fil et des types de communications sans fil. [12]

3. Les différents types de communication sans fil

3.1. Infrarouge

La communication sans fil infrarouge communique des informations dans un dispositif ou des systèmes par rayonnement IR. Le contrôle nécessite une « ligne de vision » pour fonctionner ; le récepteur doit être capable de « voir » l'émetteur à tout moment afin de recevoir des données. Les contrôles à distance par infrarouge (tels que les télécommandes universelles pour téléviseurs) sont utilisés pour envoyer des commandes à un récepteur infrarouge connecté à un microcontrôleur qui interprète ensuite ces signaux et contrôle les actions du robot. [13]

▪ **Avantages :**

- Économique.
- De simples télécommandes pour téléviseurs peuvent être utilisées comme contrôleurs.

▪ **Inconvénients :**

- Elle n'a qu'un seul canal et est sensible aux interférences lumineuses.
- La distance est limitée, la transmission infrarouge à une portée maxi de 30m sans obstacle.
- Débit de données très faible (uniquement des commandes simples)

▪ **Exemple (Thymio II)**

Thymio II (voir figure ci-dessous) est une plateforme robotique de développement Open Source complète, facile d'utilisation, idéale pour les budgets modestes!

Embarquant de nombreux capteurs et une suite logicielle complète (programmation graphique et textuelle), le robot Thymio II est une solution tout-en-un idéale pour débiter dans le développement de robots!

Thymio II est contrôlable par télécommande universelle (voir ici les télécommandes compatibles), mais il est parfois difficile de trouver la bonne. Thymio propose une télécommande infrarouge RC5, simple d'utilisation.

RC5 est un code établi par Philips qui permet de communiquer avec les différents appareils du constructeur.

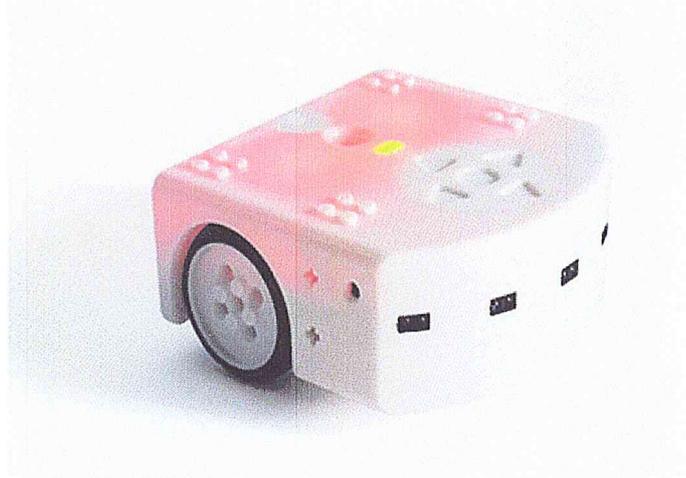


Figure II- 1: Le Robot THYMIO II.

3.2. Bluetooth

Bluetooth est une technologie sans fil qui existe depuis plus de 20 ans. Il permet à plusieurs périphériques de se connecter, d'interagir et de se synchroniser sans devoir configurer des réseaux et des mots de passe complexes. Bluetooth est partout aujourd'hui, des téléphones portables aux ordinateurs portables, et même les autoradios. Bluetooth prend en charge une grande variété de périphériques différents et peut être configuré en quelques minutes, et offre une communication bidirectionnelle.

Le Bluetooth est une forme de RF et respecte des protocoles spécifiques pour l'envoi et la réception de données. La portée Bluetooth normale est souvent limitée à environ 10m, ce qui lui procure l'avantage de permettre aux utilisateurs de contrôler leur robot via des appareils Bluetooth tels que les téléphones cellulaires, les PDA et les ordinateurs portables (même si de la programmation personnalisée peut être nécessaire pour créer une interface). [14]

▪ Avantages :

- Contrôlable à partir de n'importe quel appareil compatible Bluetooth (généralement une programmation supplémentaire est nécessaire), tel qu'un Smartphone, un ordinateur portable, un ordinateur de bureau, etc.
- Rend possible un débit de données plus élevé, le Bluetooth permet d'obtenir des débits de l'ordre de 1 Mbps, correspondant à 1600 échanges par seconde en full-duplex, avec une portée d'une dizaine de

mètres environ avec un émetteur de classe II et d'un peu moins d'une centaine de mètres avec un émetteur de classe I.

- Omnidirectionnel (n'a pas besoin de ligne de vision et peut passer un peu à travers les murs).
- Très faible consommation d'énergie, puisque la consommation qu'il engendre est extrêmement réduite.

▪ **Inconvénients :**

- Les dispositifs doivent être « jumelés ».
- L'un des gros inconvénients de Bluetooth est la sécurité. Cela est dû au fait qu'il fonctionne sur la fréquence radio et peut donc pénétrer à travers les murs. Il est conseillé de ne pas l'utiliser pour le transfert de données professionnelles ou personnelles.
- La bande passante est inférieure à celle du Wi-Fi.
- L'utilisation de la batterie est plus comparable à la condition lorsque Bluetooth est éteint. La nouvelle technologie connue sous le nom de BLE ou Bluetooth Low Energy ou Bluetooth smart est développée pour améliorer encore la durée de vie de la batterie.
- Bluetooth a une portée de seulement 15 à 30 mètres en fonction de l'appareil (sans obstacle).

▪ **Exemple :**(Robot Mbot Blue STEM Bluetooth de Makeblock)

Cette version contient un programme optimisé du Mbot V1.1 et les utilisateurs peuvent vérifier si Mbot est allumé juste en utilisant le bouton embarqué. Le robot possède les mêmes caractéristiques de l'ancien modèle mais aussi des modifications supplémentaires.

De plus, contrairement à l'ancien modèle, ce modèle possède une coque supplémentaire qui protège le panneau principal de contrôle. La boîte du robot est également conçue pour protéger les yeux des enfants de la luminosité de la LED sur le robot.

Les roues auxiliaires sont en nylon ce qui permet un changement de direction plus facile et qui réduit le petit claquement métallique de la version précédente. Comme le modèle V1.1, le robot est programmable via Open source avec Scratch 2.0 ou Arduino. Sa programmation est facile et peut s'effectuer avec

des blocks. La connexion sans fil s'effectue via Bluetooth. Dans la boîte, est également offerte une télécommande pour contrôler le robot.

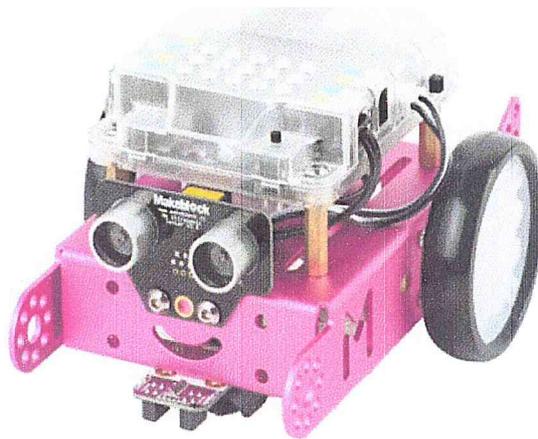


Figure II- 2: Le Robot Mbot Blue STEM.

3.3. Wi-Fi

Le Wi-Fi, est un ensemble de protocoles de communication sans fil régis par les normes du groupe IEEE 802.11. La portée peut atteindre plusieurs dizaines de mètres en intérieur (entre 20m jusqu'à 50m) s'il n'y a aucun obstacle gênant entre l'émetteur et le récepteur, permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, Smartphone, modem Internet, etc.) au sein d'un réseau informatique afin de permettre la transmission de données entre eux.

Le Wi-Fi est à présent une option pour les robots ; être capable de contrôler un robot sans fil via l'Internet présente des avantages importants (et quelques inconvénients) pour du contrôle sans fil. Afin de mettre en place un robot Wi-Fi, on a besoin d'un routeur sans fil connecté à Internet et d'une unité Wi-Fi sur le robot lui-même. Pour le robot, on peut également utiliser un dispositif acceptant le TCP/IP avec un routeur sans fil. [15]

▪ **Avantages :**

- **Mobilité:** La connexion au réseau sans fil permet de se déplacer librement dans le rayon disponible. On peut ainsi emmener son laptop de la salle de réunion à l'atelier sans avoir à brancher/débrancher quoi que ce soit.

- **Facilité:** Un réseau Wi-Fi bien configuré permet de se connecter très facilement, à condition, bien sûr, de posséder une autorisation. Il suffit généralement de se trouver dans la zone de couverture pour être connecté.
 - **Souplesse:** La souplesse d'installation du Wi-Fi permet d'adapter facilement la zone d'action en fonction des besoins. Si le point d'accès est trop faible, on ajoute des répéteurs pour étendre la couverture.
 - **Coût:** La plupart des éléments du réseau Wi-Fi (point d'accès, répéteurs, antennes...) peuvent être simplement posés. L'installation peut donc parfois se faire sans le moindre outillage, ce qui réduit les coûts de main-d'œuvre. Le budget de fonctionnement est similaire à un réseau filaire.
 - **Evolutivité:** La facilité d'extension ou de restriction du réseau permet d'avoir toujours une couverture Wi-Fi correspondant aux besoins réels.
- **Inconvénients :**
 - **Qualité et continuité du signal:** Un réseau Wi-Fi bien installé et bien configuré est généralement fiable et d'une qualité constante. Cependant, il suffit parfois de peu pour perturber le signal : un radar de gendarmerie ou un émetteur Bluetooth, par exemple.
 - **Sécurité:** Le Wi-Fi étant un réseau sans fil, il est possible de s'y connecter sans intervention matérielle. Cela veut dire qu'il faut particulièrement étudier la sécurisation du réseau si l'on veut éviter la présence d'indésirables ou la fuite d'informations

- **Exemple (Kompai)**

Kompai est un robot d'assistance aux personnes âgées, qui peut faire de la téléprésence. C'est-à-dire qu'un opérateur peut prendre le contrôle du robot à distance, et déclencher une Visio conférence pour faire de la levée de doute, mais également déplacer le robot dans son environnement pour rechercher la personne si celle-ci ne décroche pas. Pour cela, il a à sa disposition une



Figure II- 3: Le robot Kompai.

- Il faut connaître le système pour que le propriétaire puisse utiliser des dispositifs conformes au zigbee.
- Il n'est pas sécurisé comme un système sécurisé basé sur le wifi.
- Comme les autres systèmes sans fil, la communication basée sur zigbee est sujette aux attaques de personnes non autorisées.

▪ **Exemple :**(ZigBee Controlled Pick & Place)

Le robot Pick & Place ZigBee est composé d'un robot à roues dentées équipé d'un module de préhension. Ce robot peut ramasser des objets à l'aide d'un module de préhension à partir d'un seul endroit, effectuer toutes sortes de mouvements robotiques et placer les produits cueillis à un autre endroit. Le robot est interfacé avec un module ZigBee du côté de l'entrée ainsi que du côté robotique. Cela initie à fournir des commandes d'entrée de l'utilisateur à l'extrémité du récepteur où les actions robotiques sont effectuées.

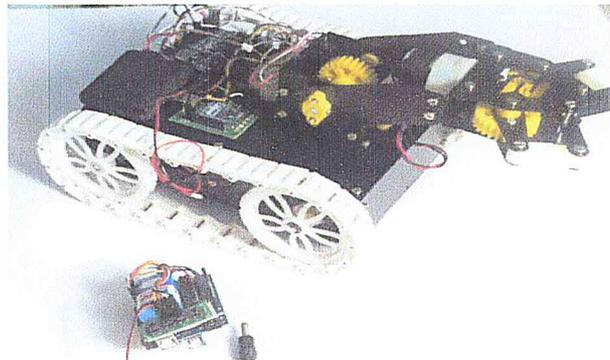


Figure II- 4: Le robot (ZigBee Controlled PICK & PLACE).

4. Choix de la technologie radio pour la connectivité sans fil des objets utilisés

Après avoir fait état dans la section précédente sur les importants critères pour le choix de la technologie radio, dans cette section nous allons faire une comparaison des performances des différentes normes de communication sans-fil pour chacun de ces critères.

4.1. Distance maximale (portée) :

La distance maximale est déterminée par le type de technologie radio, elle est liée à la puissance de sortie, la sensibilité du récepteur et la fréquence.

Communication en champ proche appelée aussi la norme NFC (Near Field Communication) est une technologie dédiée aux courtes distances, avec une distance typique d'environ 5cm. Le Zigbee réalise la meilleure portée (jusqu'à 1,5 km en plein air et en visée directe) grâce à une puissance de sortie élevée (jusqu'à 18dBm) et une bonne immunité aux interférences. Ce chiffre reste théorique, d'autant plus qu'en France la norme limite les émissions à une puissance de 10dBm, ce qui réduit la portée maximale à environ 750m.

Actuellement la norme Wi-Fi 802.11y permet d'atteindre une portée de 5 km. Cependant, cette évolution est réservée à la bande 3.7GHz qui ne sera utilisée qu'aux États-Unis. De plus c'est une norme qui n'est pas dédiée aux objets communicants eux-mêmes, mais plutôt à l'interconnexion de réseaux locaux.

La figure (II-5) montre une classification des différentes technologies en fonction de leurs distances maximales. [17]

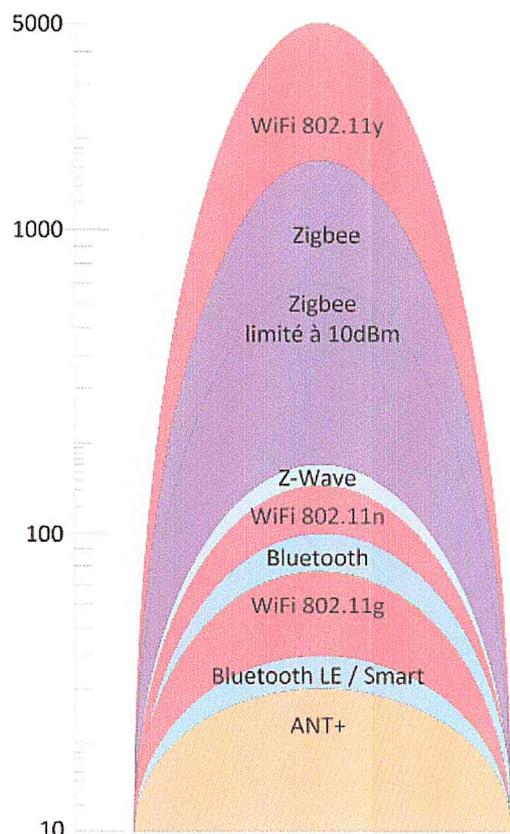


Figure II- 5: Portée comparée (en mètres) des principaux standards de connectivité sans fil.

4.2. Débit :

C'est la quantité de données qui peuvent être transmises d'un point à un autre en un laps de temps déterminé. Il détermine la vitesse de transmission des informations sur un réseau informatique.

Pour garantir la transmission d'une grande quantité de données avec un temps de transmission réduit et un minimum de consommation d'énergie, il est important de choisir une technologie aux débits élevés.

La figure suivante montre un graphe qui présente une comparaison entre les différentes technologies radio en fonction de la distance maximale et du débit : [17]

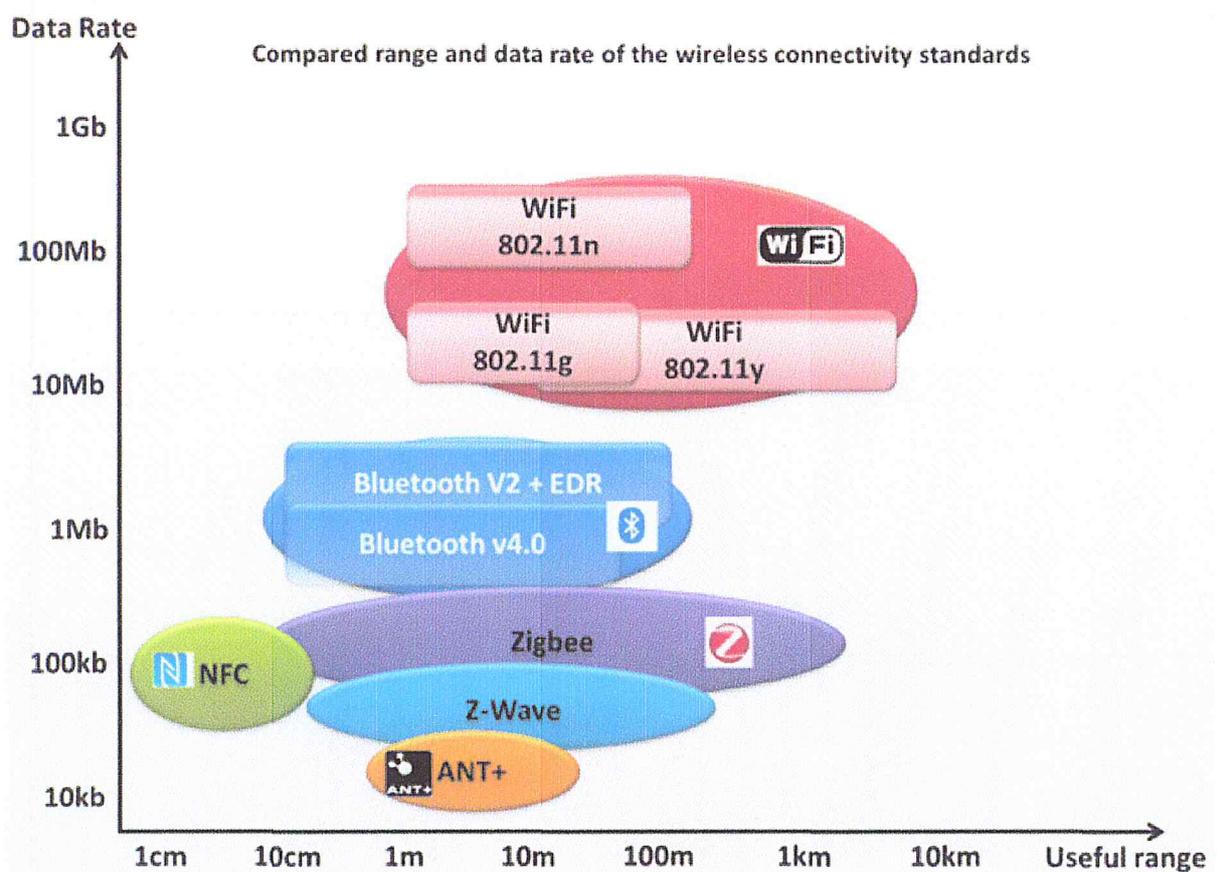


Figure II- 6: Comparaison de la portée et du débit (en bits/s) des principaux standards de connectivité sans fil.

4.3. Autonomie

Le choix de la technologie radio détermine l'autonomie maximale. Egalement, l'énergie consommée par un objet pour transmettre et recevoir des données et sa capacité à ne rien consommer quand il n'y a rien à transmettre ou à recevoir déterminent l'autonomie de cet objet. C'est la notion d'endormissement.

Le débit maximum impacte la consommation. En effet, S'il y a beaucoup de données à transmettre, il faut pouvoir les transmettre rapidement pour pouvoir s'endormir ensuite plus longtemps. Ainsi il vaut mieux consommer un peu d'énergie de temps en temps et ne rien consommer le reste du temps que consommer moins d'énergie, mais tout le temps.

Pour évaluer la consommation, on tiendra donc compte du type d'utilisation, et on calculera une consommation moyenne dans le temps.

Lorsque la consommation moyenne de courant augmente, soit l'autonomie se dégrade, soit il faut augmenter la taille de la batterie.

La figure (II-7) présente une comparaison de la consommation d'énergie des différentes technologies de la connectivité sans fil. [17]



Figure II- 7: Comparaison de la consommation d'énergie des principaux standards de connectivité sans fil.

Après cette série de comparaisons entre différents commande à distance, nous constatons que le Wifi est la solution la plus appropriée à notre système vu ces multiples caractéristiques positives.

5. Conclusion

Ce chapitre est une présentation des commandes à distances qui existent dans un robot mobile, la communication sans fil a été étudiée, comment cela fonctionne, les avantages et les inconvénients de chaque type de commande.

Nous avons présenté la relation entre diverses technologies et comment celles-ci ont évolué au fil du temps.

Dans le prochain chapitre nous présentons notre solution ainsi que les diverses technologies qu'elle utilise.

Chapitre III:

Conception

1. Introduction

Dans ce chapitre nous présentons la partie conception en commençant par décrire les besoins fonctionnels et non fonctionnels de notre étude ainsi que le formalisme d'UML que nous avons choisi comme langage de modélisation pour nous aider à structurer notre travail.

2. Spécification des besoins

Il existe deux types des besoins :

2.1. Besoins fonctionnels

Pour assurer les différents besoins et attentes des utilisateurs, notre système devra être extensible et performant permettant de manipuler et contrôler le robot et récupérer le flux vidéo.

Notre système comprendra trois (03) fonctionnalités principales que nous allons présenter ci-dessous de façon plus détaillée afin d'être plus clairs.

2.1.1. Gestion des utilisateurs:

On distingue principalement deux types d'utilisateurs : "administrateur" et "client" pour la plateforme de gestion dont le premier permet à l'utilisateur d'accéder au site et créer les différents clients et autoriser ou interdire à l'utilisateur d'accéder au site pour contrôler le robot.

2.1.2. Vidéo surveillance:

La vidéo surveillance doit permettre à l'utilisateur de surveiller son lieu en ligne avec possibilité d'enregistrement des vidéos, et en particulier celles de guider le robot jusqu'aux endroits souhaités, en offrant à l'utilisateur :

- Un accès à distance.
- Échange des flux vidéo des deux côtés.

2.1.3. Manipulation du robot:

Contrôle à distance du robot et gestion dans toutes les directions souhaitées (avancer, reculer, tourner) grâce à notre site, en envoyant certaines valeurs de vitesse au robot distant, et le robot les convertit en mouvement.

2.2. Besoins non fonctionnels

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate aux attentes des utilisateurs. En plus de l'économie de l'énergie qui est le besoin principal, notre application doit nécessairement assurer ce qui suit :

- L'extensibilité : dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- La sécurité : l'application devra être hautement sécurisée, les informations ne devront pas être accessibles à tout le monde, c'est-à-dire que le site web est accessible par un identifiant et un mot de passe attribué à une personne physique.
- L'interface: avoir une application qui respecte les principes des interfaces Homme/Machine (IHM) tels que l'ergonomie et la fiabilité.
- La performance: l'application devra être performante c'est-à-dire que le système doit réagir dans un délai précis, quel que soit l'action de l'utilisateur
- La convivialité : l'application doit être simple et facile à manipuler même par des non experts.
- L'ergonomie : le thème adopté par l'application doit être clair et donner une inspiration futuriste.

3. Architecture du système

L'architecture de notre système est composée des éléments suivants :

- Un utilisateur (Pc, Laptop, Tablette, Smartphone...).
- Un robot composé d'une caméra, une tablette, et un PC embarqué.
- Une base de données pour identifier les utilisateurs.
- Un serveur pour gérer la communication entre le robot et l'utilisateur.
- Les composants (utilisateur, robot, serveur) communiquent à travers une liaison Wifi

La figure suivante représente le fonctionnement des composants de l'architecture de notre système :

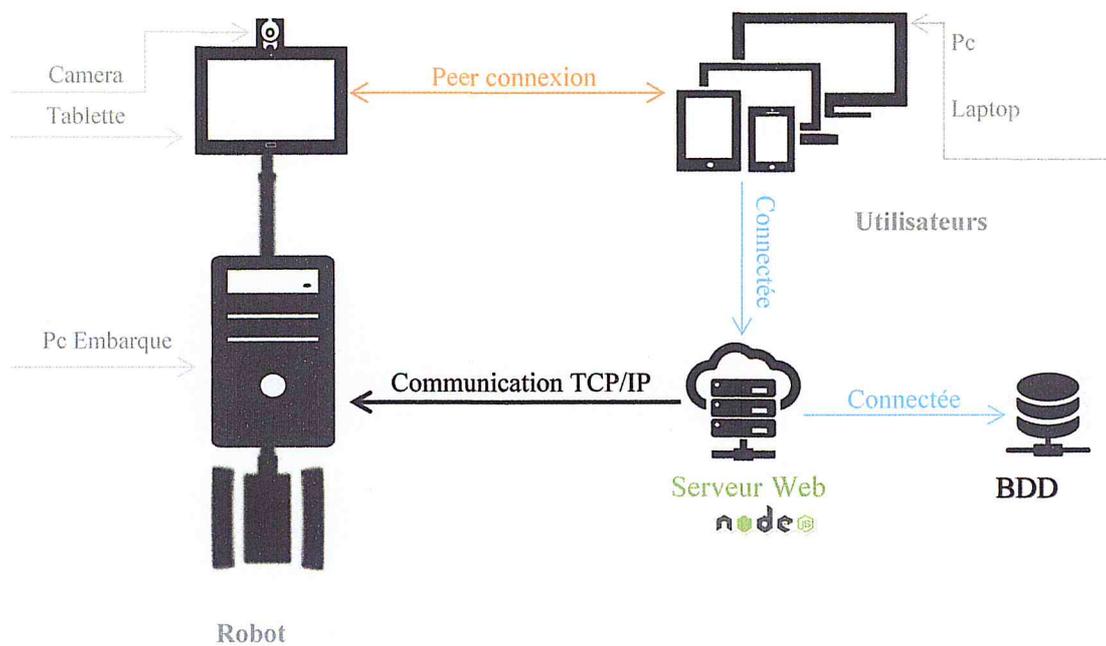


Figure III - 1: Architecture générale du système.

Afin de faciliter le développement de notre système, on propose de diviser la topologie de notre système en deux parties:

3.1. Communication par vidéo (Media)

La figure suivante représente l'architecture de communication par vidéo :

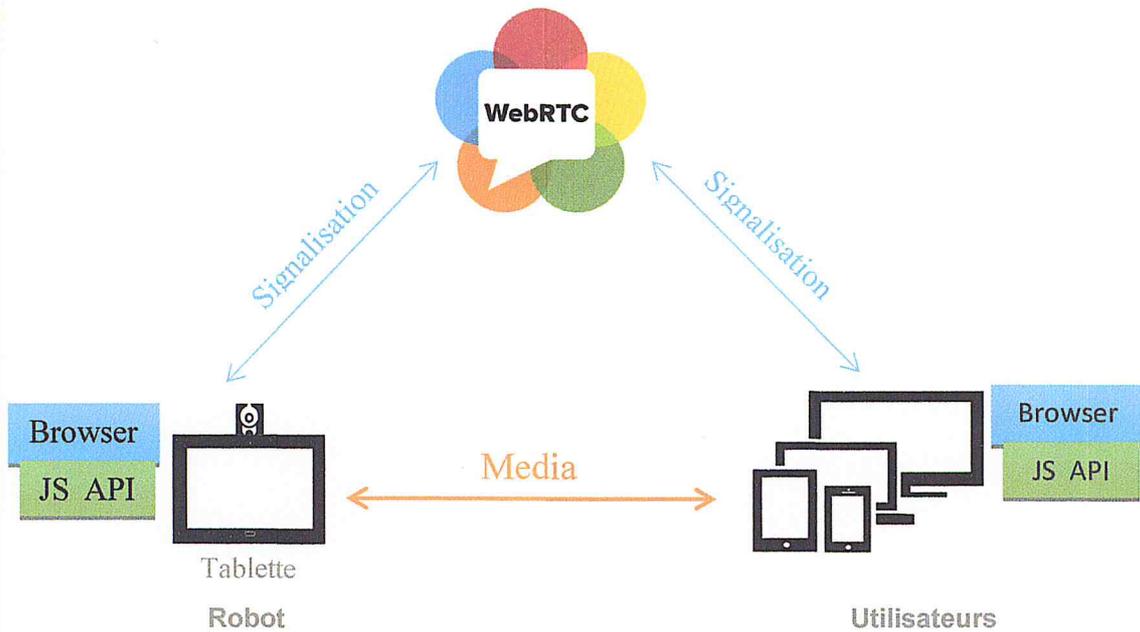


Figure III - 2: Architecture de communication par vidéo (Media).

Nous avons choisi de faire une communication media entre deux navigateurs web (L'utilisateur et le Robot). Pour faciliter cette communication on a utilisé la technologie WebRTC à cause de leur rapidité de communication entre les navigateurs web. Si l'utilisateur veut démarrer une communication avec le Robot, on va générer une offre, cette offre à un format bien particulier, qui contient des informations concernant l'utilisateur, par exemple le chemin et son adresse globalement. Après la génération de cette offre, l'utilisateur va la transmettre au robot via le serveur web. Une fois le robot reçoit cette offre, lui aussi va générer une offre qui contient des informations, et va envoyer à l'utilisateur de la même façon. Une fois que l'utilisateur et le robot ont leurs offres mutuelles ils peuvent commencer une communication en Peer to Peer.

3.2. Communication par des messages (Data)

La figure suivante représente la communication par des messages (Data), et les langages utilisés dans le système :

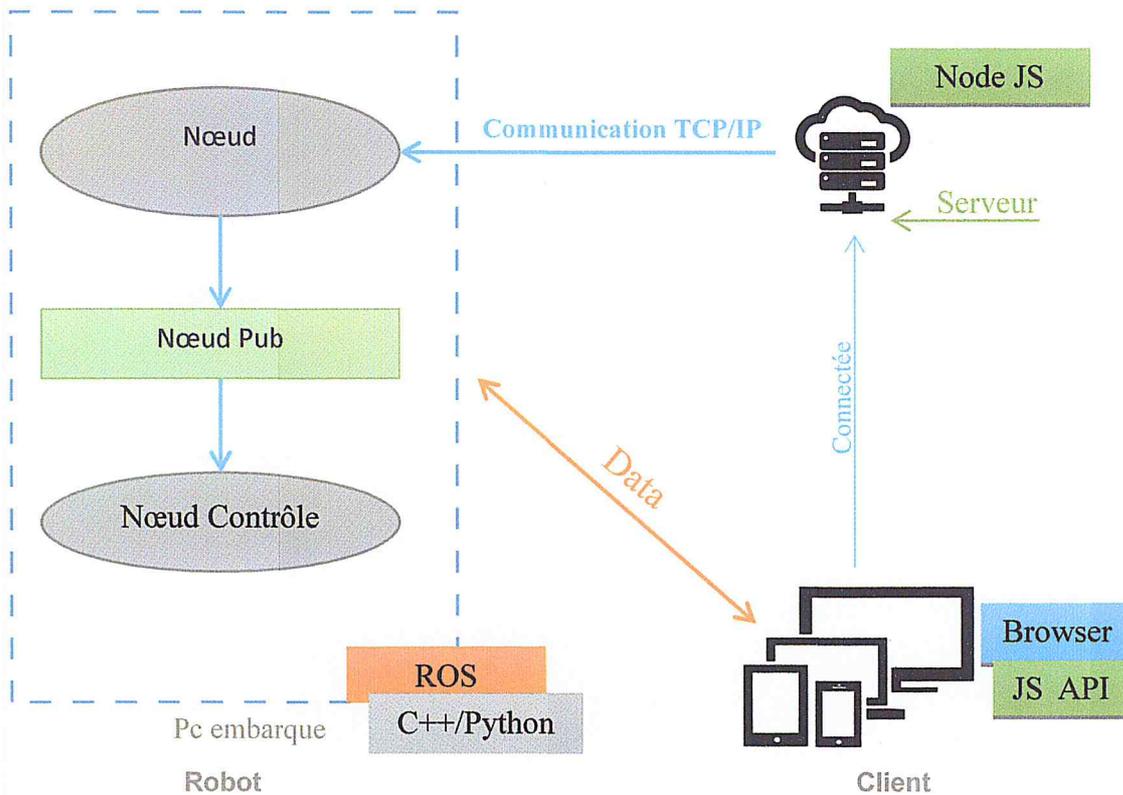


Figure III - 3: Architecture de communication par des messages (Data).

Pour l'échange des données on a utilisé le protocole TCP/IP comme un protocole de transfert, puisque on n'a pas besoin de grande bande passante, et il est nécessaire de garantir la sécurité.

Le client envoie des commandes au serveur (contient la vitesse angulaire et la vitesse linéaire), ensuite le serveur va transmettre les vitesses au nœud de communication, ce dernier va enregistrer chaque vitesse reçue dans un fichier texte. Le nœud Pub va lire le fichier ".txt" et transférer les vitesses au nœud de contrôle qui va manipuler le robot.

4. Modélisation

Chaque application et chaque système avant d'être réalisé doit passer par une étape de conception avec une méthode ou bien un langage de modélisation. Cette étape permet de décrire les fonctionnalités du système, son comportement et tout le détail nécessaire pour la réalisation de ce système et son déroulement.

Les langages de modélisation orientés objet ont fait leur apparition entre le milieu des années soixante-dix et la fin des années quatre-vingt, quand les spécialistes du génie logiciel étaient confrontés à un nouveau genre de langages de programmation orientés objet, et à des applications de plus en plus complexes.

La conception de l'application vise principalement à préciser le modèle d'analyse de telle sorte qu'il peut être implémenté avec les composants de l'architecture. Cette opération représente la phase la plus complexe du projet.

4.1. Définition d'UML

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. [18]

4.2. Les différents types de diagrammes UML

UML propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles.

- **Les diagrammes structurels** : Ces diagrammes, au nombre de six, ont vocation à représenter l'aspect statique d'un système (classes, objets, composants...). [19]
 - Diagramme de classes : Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.
 - Diagramme d'objets : Il montre les instances des éléments structurels et leurs liens à l'exécution.
 - Diagramme de packages : Il montre l'organisation logique du modèle et les relations entre packages.
 - Diagramme de structure composite : Il montre l'organisation interne d'un élément statique complexe.
 - Diagramme de composants: Il montre des structures complexes, avec leurs interfaces fournies et requises.
 - Diagramme de déploiement: Il montre le déploiement physique des « artefacts » sur les ressources matérielles.

- **Les diagrammes de comportement** : Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML: [19]
 - Diagramme de cas d'utilisation : Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.
 - Diagramme de vue d'ensemble des interactions : Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.
 - Diagramme de séquence : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.
 - Diagramme de communication : Il montre la communication entre objets dans le plan au sein d'une interaction.
 - Diagramme de temps : Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.
 - Diagramme d'activité : Il montre l'enchaînement des actions et décisions au sein d'une activité.
 - Diagramme d'états : Il montre les différents états et transitions possibles des objets d'une classe.

4.3. Application du langage de modélisation UML à notre étude

Après avoir défini le langage UML et ses diagrammes, nous avons identifié les acteurs et choisi les trois diagrammes pour notre étude :

- Diagramme de cas d'utilisation.
- Diagramme de séquence.
- Diagramme de classe.

4.3.1. Définition d'un acteur

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Dans notre cas deux types d'acteurs seront définis pour utiliser notre application :

- **Administrateur** : l'administrateur est le responsable de la maintenance et la mise à jour de l'application. Il a des fonctionnalités supplémentaires qu'un utilisateur simple, ainsi que la possibilité d'ajouter un nouvel utilisateur et de modifier ou activer/désactiver un ancien utilisateur du système.
- **Utilisateur** : peut accéder à des fonctionnalités attribuées par l'administrateur du système (Récupérer les flux vidéo, manipuler le robot, ...).
- **Robot** : le rôle du robot est de recevoir les commandes d'un utilisateur du système et de transformer un mouvement, et attendre une connexion à un moment donné.

4.3.2. Les cas d'utilisations:

Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système produisant un service ou une valeur ajoutée pour un acteur particulier. Il exprime les interactions (acteur/système) et permet de décrire ce que le futur système devra faire, sans spécifier le comment. [21] L'objectif est le suivant : L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un de ses acteurs voir III-4.

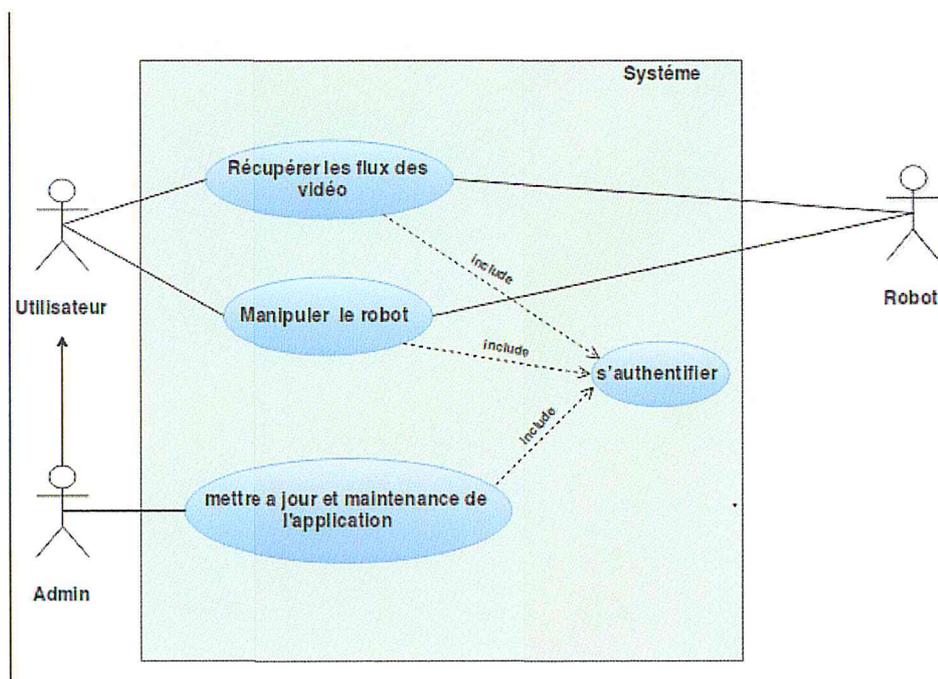


Figure III -4: Diagramme général des cas d'utilisations.

▪ Description textuelle de cas d'utilisation « S'authentifier »:

Titre	S'authentifier
Acteur	Utilisateur
Description	Avoir accès au système.
Précondition	Avoir déjà un compte.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur saisit son nom d'utilisateur et son mot de passe. 2. Le système vérifie le nom et le mot de passe. 3. Le système autorise l'accès à l'utilisateur.
Scénario alternatif	<ul style="list-style-type: none"> ➤ Si un champ n'est pas rempli alors un message est affiché « veuillez renseigner ce champ ». ➤ Si le nom d'utilisateur ou le mot de passe sont incorrects, un message est affiché « le nom ou mot de passe incorrects ».
Post-condition	L'utilisateur est connecté au système, et redirigé vers la page principale de l'application.

Tableau III -1: Description textuelle du cas d'utilisation « s'authentifier ».

- Maintenant nous allons détailler ce diagramme général en détaillant les différents cas d'utilisation en commençant par le 1er cas d'utilisation qui est Récupérer le flux vidéo.

4.3.2.1. Cas d'utilisation « Récupérer le flux vidéo »

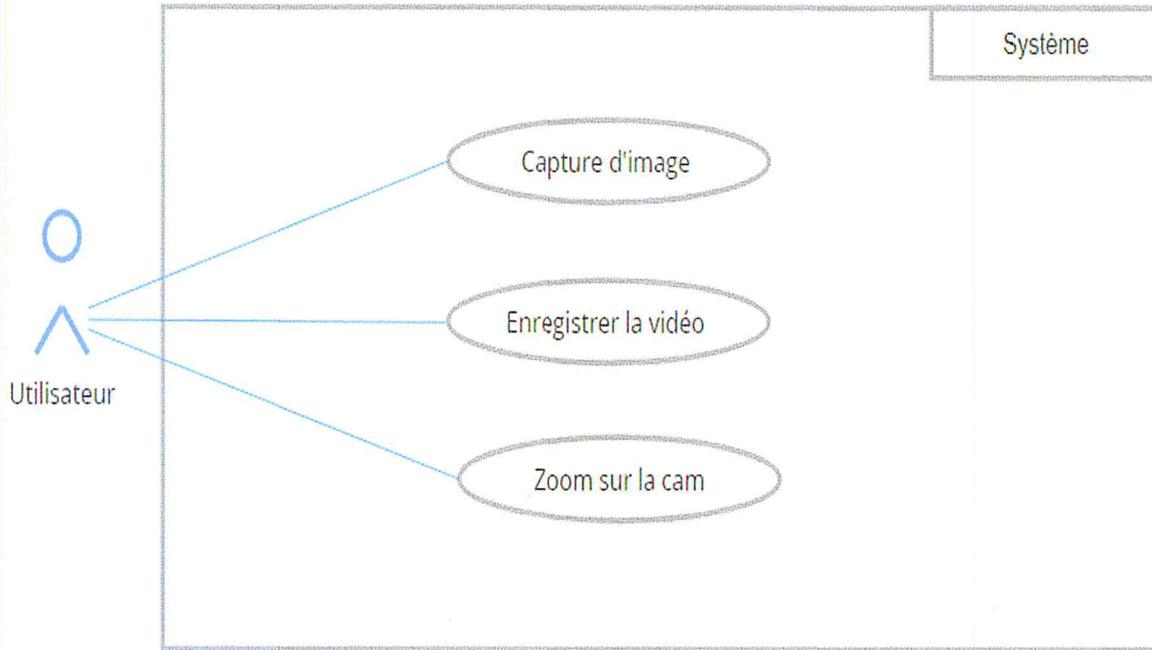


Figure III - 5: Diagramme de cas d'utilisation « Récupérer le flux vidéo ».

▪ Description textuelle de cas d'utilisation « Récupérer le flux vidéo »:

Titre	Récupérer le flux vidéo
Acteur	Utilisateur
Description	Avoir accès à la caméra
Précondition	Utilisateur doit être connecté au système
Scénario nominal	1. L'utilisateur lance un appel vidéo au robot. 2. La caméra du robot accepte l'appel. 3. L'utilisateur récupère le flux vidéo.
Post-condition	L'utilisateur est connecté au robot et reçoit le flux vidéo.

Tableau III -2:Description textuelle du cas d'utilisation « Récupérer le flux vidéo »

4.3.2.2. Cas d'utilisation « Manipuler le robot »

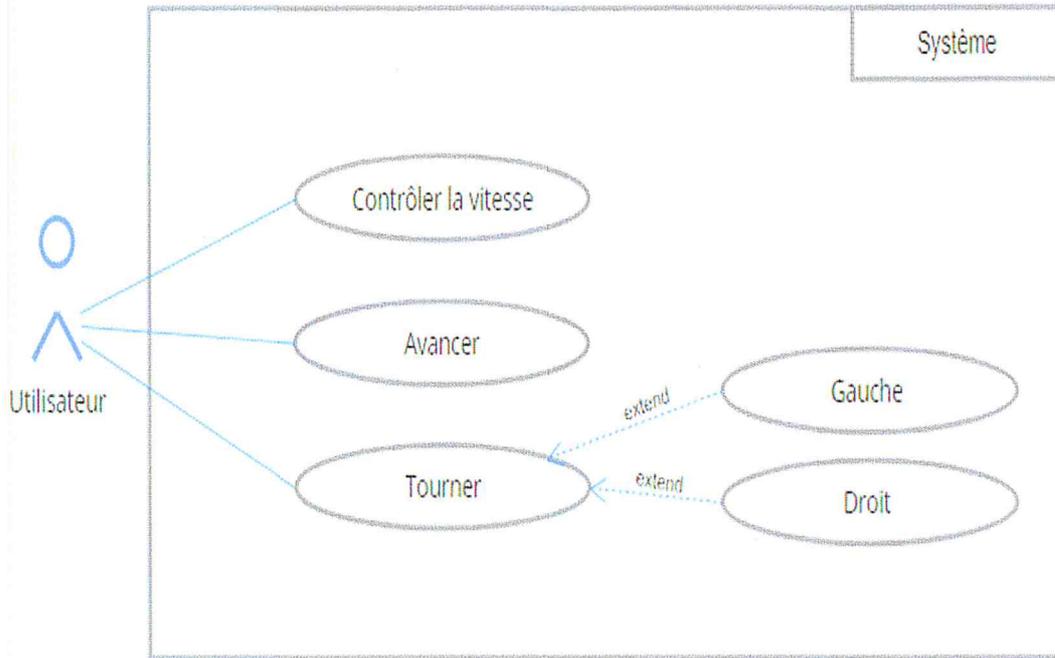


Figure III - 6: Diagramme du cas d'utilisation « Manipuler le robot ».

▪ Description textuelle de cas d'utilisation « Manipuler le robot »:

Titre	Manipuler le robot
Acteur	Utilisateur
Description	Manipulation du robot à distance
Précondition	Utilisateur doit être connecté au système
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur ouvre la page « Accueil » 2. L'utilisateur clique sur le bouton pour avancer. 3. Le système envoie les données au robot. 4. Le robot reçoit les données et les transforme en mouvement.

Tableau III -3: Description textuelle du cas d'utilisation « manipuler le robot »

4.3.2.3. Cas d'utilisation « Mettre à jour et maintenance d'application »

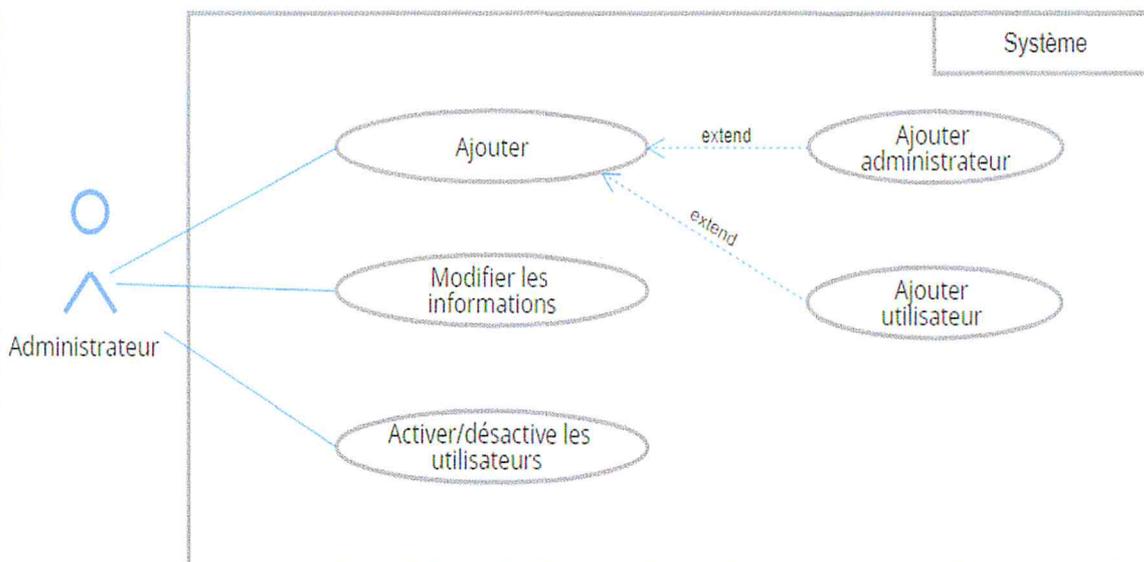


Figure III - 7: Diagramme du cas d'utilisation « Mettre à jour et maintenance d'application ».

▪ Description textuelle du cas d'utilisation « Modifier les informations »:

Titre	Modifier les informations
Acteur	Administrateur
Description	Mettre à jour les informations de l'utilisateur.
Précondition	Administrateur doit être connecté au système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur ouvre la page « Paramètres ». 2. L'administrateur change les informations des utilisateurs. 3. Le système met à jour les informations de l'utilisateur.
Scénario alternatif	<p>Selon ce qu'il a rempli comme champ, alors un message lui est affiché « Veuillez renseigner tel ou tel champ ».</p> <ul style="list-style-type: none"> ➤ Si le nom d'utilisateur ou le mot de passe sont incorrects, un message est affiché « le nom ou mot de passe incorrects ». ➤ Sinon, un message affiche « la modification est valide »
Post-condition	Profil utilisateur mis à jour.

Tableau III -4: Description textuelle du cas d'utilisation « Modifier les informations ».

▪ **Description textuelle de cas d'utilisation « Activer/désactiver les utilisateurs »:**

Titre	Activer/désactiver les utilisateurs
Acteur	Administrateur
Description	Activer/désactiver les utilisateurs
Précondition	Administrateur doit être connecté au système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur ouvre la page 'paramètres'. 2. L'administrateur active/désactive les profils. 3. Un message est affiché « vous êtes sur le point d'activer/désactiver le profil ». 4. Si confirmation, le Profil sera activé/désactivé.
Post-condition	Profil activé ou désactivé.

Tableau III -5: Description textuelle du cas d'utilisation « Activer/désactiver les utilisateurs ».

▪ **Description textuelle du cas d'utilisation « Ajouter utilisateur »:**

Titre	Ajouter utilisateur
Acteur	Administrateur
Description	Ajouter un utilisateur au système
Précondition	Administrateur doit être connecté au système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur ouvre la page 'paramètres' et lance une demande d'ajout 2. Un formulaire d'inscription est affiché. 3. L'administrateur saisit le nom d'utilisateur et le mot de passe d'utilisateur. 4. L'inscription est effectuée et l'utilisateur est ajouté.
Scénario alternatif	<p>Selon ce qu'il a rempli comme champ, alors un message lui est affiché « Veuillez renseigner tel ou tel champ ».</p> <ul style="list-style-type: none"> ➤ Si le nom d'utilisateur ou le mot de passe sont incorrects, un message est affiché « le nom ou mot de passe incorrects ». ➤ Sinon, un message affiche « L'inscription est valide ».
Post-condition	Ajouter un nouvel utilisateur.

Tableau III -6: Description textuelle de cas d'utilisation « Ajouter utilisateur ».

4.3.4. Diagrammes de séquence

Un diagramme de séquence est un moyen de capturer le comportement de tous les objets et les acteurs impliqués dans un cas d'utilisation.

4.3.3.1. Diagramme de séquence du cas d'utilisation « Tourner le robot »

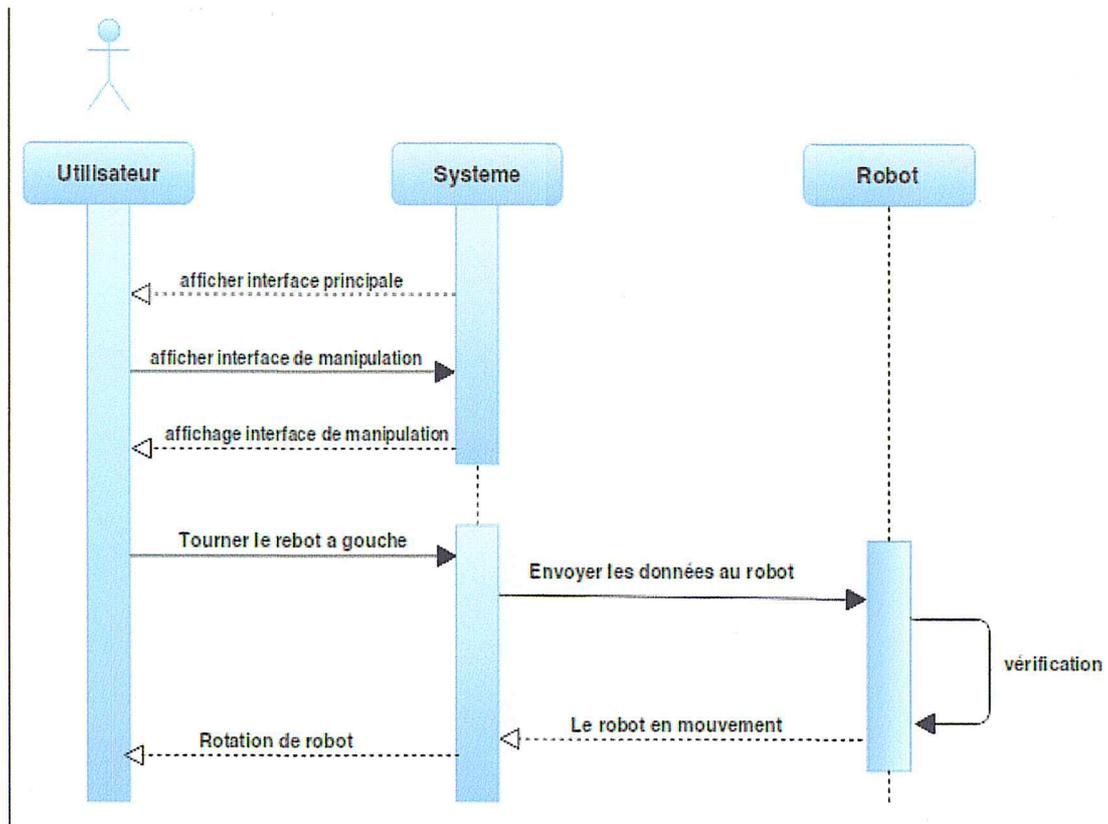


Figure III - 8: Diagramme de séquence « Tourner le robot ».

4.3.3.2. Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur »

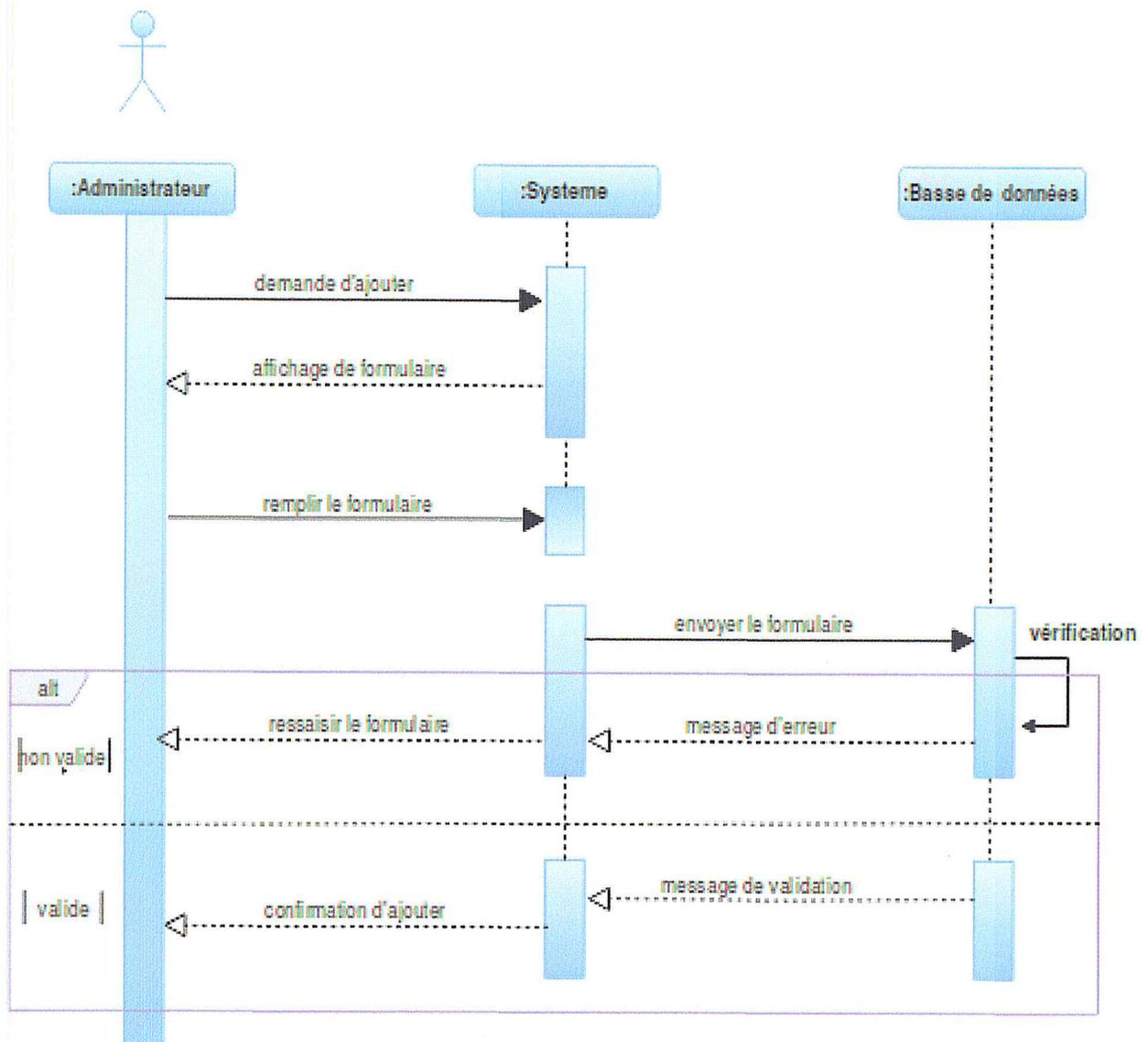


Figure III - 9: Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur ».

4.3.3.3. Diagramme de séquence du cas d'utilisation « Modifier un utilisateur »

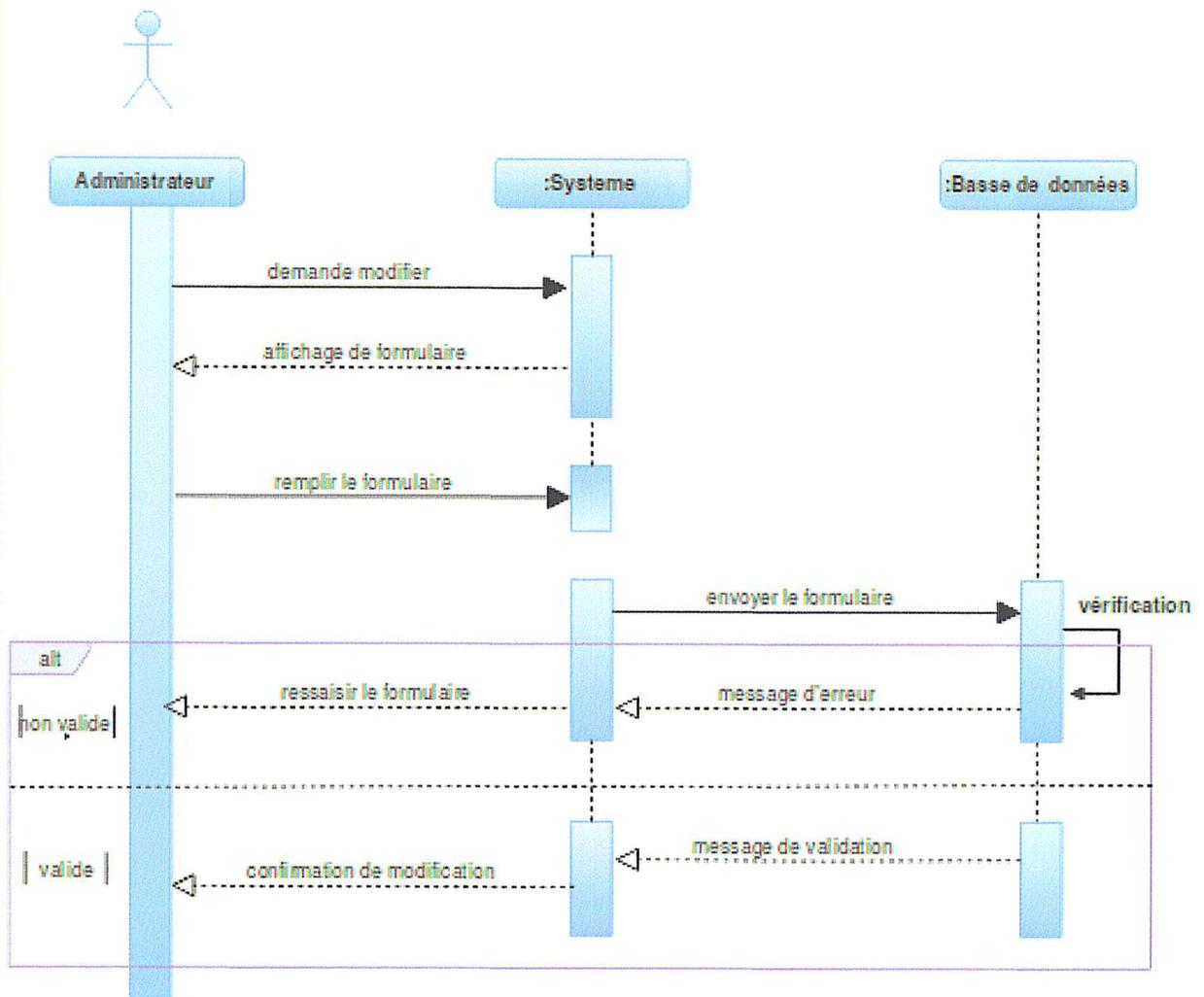


Figure III - 10: Diagramme de séquence du cas d'utilisation « Modifier un utilisateur ».

4.3.4. Diagramme de classe

Les diagrammes de classes décrivent les types des objets qui composent un système et les différents types de relations statiques qui existent entre eux. Les diagrammes de classes font abstraction du comportement du système.

Les classes qui composent notre système sont :

- Classe utilisateur
- Classe Administrateur
- Classe Robot

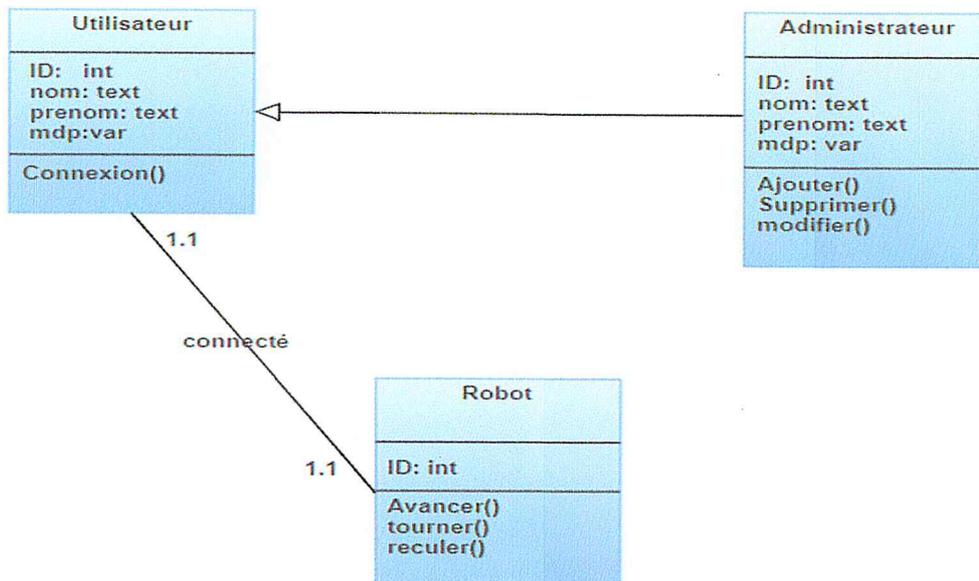


Figure III - 11: Diagramme de classes.

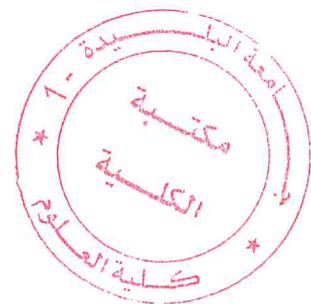
- Ajouter () : Permet d'ajouter un utilisateur.
- Supprimer () : Permet de supprimer un utilisateur.
- Modifier () : Permet de modifier un utilisateur.
- Connexion () : Permet d'établir une connexion avec le système.
- Avancer () : Permet de faire avancer le robot.
- Tourner () : Permet de tourner le robot.
- Reculer () : Permet de reculer le robot.

5. Conclusion

A travers ce chapitre, nous avons présenté la conception de l'application, et avons défini les différents besoins fonctionnels et non fonctionnels et l'architecture de notre système. Ensuite, nous avons présenté la conception logicielle de l'application à travers les diagrammes de cas d'utilisation, les diagrammes de séquences et le diagramme de classes.

Dans le chapitre suivant, nous allons passer à l'étape de la réalisation de notre application.

Chapitre IV : Implémentation



1. Introduction

Après avoir complété l'étude conceptuelle dans le chapitre précédent, nous présentons la partie réalisation de notre application. Nous procédons dans ce chapitre par une description de l'environnement logiciel de développement, et nous terminons ce chapitre par présenter et décrire quelques interfaces de notre application.

▪ A propose de Robot

Le robot mobile B21r est une plate-forme expérimentale construite par la société iRobot (fig. IV-1) pouvant se déplacer sur un terrain non accidenté ayant comme type de traction, la traction synchrone. Il dispose de quatre roues décentrées orientables tournant selon deux axes : une rotation selon l'axe y pour engendrer la translation, et une rotation selon l'axe verticale au sol pour engendrer une rotation sur lui-même. Ce robot mobile est muni de deux ceintures de capteurs à ultrasons, une ceinture de capteurs infrarouges, un laser, des capteurs tactiles placés le long de ses parois, et une caméra CCD N/B.

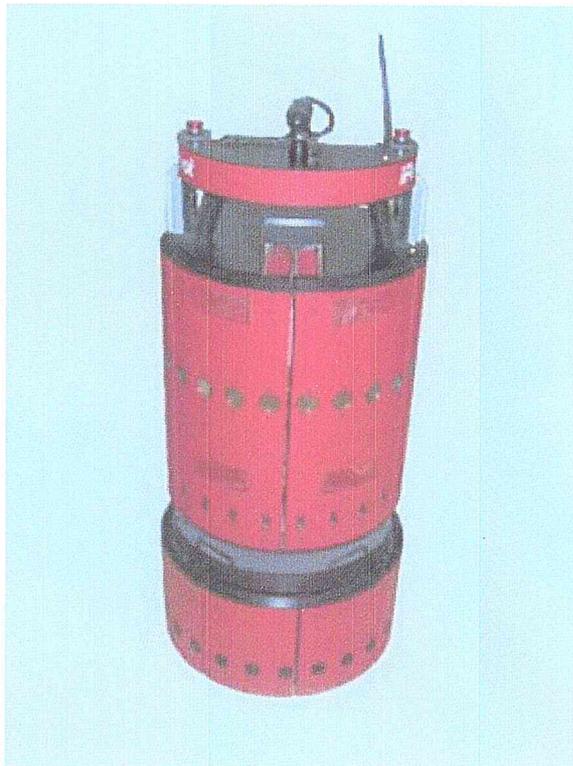


Figure IV- 1: Structure générale d'un robot

2. Outils et langages utilisés

2.1.MySQL

Le plus populaire des serveurs de bases de données SQL Open Source, c'est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, en concurrence avec Oracle, Informix et Microsoft SQL Server.

2.2.JavaScript

Le JavaScript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et pas au niveau du serveur web. [22]

2.3.Python

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités, qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées qui aident le développeur à travailler sur des projets particuliers. On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. [23]

2.4. HTML5

HTML5 (Hypertext Markup Language revision 5), est un langage de balisage pour la structure et la présentation des contenus Web. HTML5 prend en charge la syntaxe traditionnelle de style HTML et XHTML et d'autres nouvelles fonctionnalités dans son balisage, ses nouvelles API, son langage XHTML et sa gestion des erreurs.

2.5. CSS

CSS, ou feuilles de styles en cascade, a été créé en 1996 et a pour rôle de mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelées fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML. Ils contiennent par exemple du code permettant de définir la taille, la couleur ou l'alignement d'un texte.

3. Implémentation

L'implémentation de notre application web se déroule en deux grandes étapes :

- Mise en place du serveur Node.js.
- Mise en place du serveur ROS.

3.1. Server Node.js

3.1.1. Node.js

- **Définition**

Node.js est une plateforme logicielle libre et événementielle en JavaScript pour construire facilement des applications de réseau rapides et évolutives. Elle utilise la machine virtuelle V8 de Google Chrome. Node.js contient une bibliothèque de serveur HTTP intégrée, ce qui rend possible de construire ou de créer un serveur web sans avoir besoin d'un logiciel externe comme Apache ou autres. Il permet aussi de mieux contrôler la façon dont le serveur web fonctionne, tout en étant idéal pour les applications en temps réel intensives en données qui courent à travers des dispositifs distribués.

- **Principe de fonctionnement**

JavaScript avait toujours été utilisé du côté du client, c'est-à-dire l'utilisateur qui navigue sur un navigateur web comme Chrome, Opera ou autre ... Ce navigateur exécute du code JavaScript qui effectue des changements visibles que pour ce navigateur.

Contrairement à PHP qui s'exécute du côté du serveur, Node.js offre un environnement côté serveur qui permet d'utiliser le langage JavaScript pour la

génération des pages web. En résumé, Node.js vient comme remplaçant moderne et technologique de PHP ou de J2EE.

Comme JavaScript est un langage basé sur les événements, Node.js est aussi basé sur les événements. Donc, c'est toute la façon d'écrire des applications web qui change. Et c'est de la que Node.js tire toute sa puissance et sa rapidité. [24]

▪ L'extensibilité

Il est intéressant de signaler que le noyau de Node.js est tout petit. A la base, Node.js ne sait en fait pas faire grand-chose. Pourtant, Node.js est très riche grâce à son extensibilité. Ces extensions de Node.js sont appelées des « modules ».

Il existe des milliers de modules qui offrent des fonctionnalités variées : de la gestion des fichiers uploadés à la connexion aux bases de données MySQL ou à Redis, en passant par des Framework, des systèmes de Template et la gestion de la communication temps réel avec les utilisateurs ou clients. Il existe ainsi une panoplie de modules et de nouveaux apparaissent chaque jour. [25]

3.1.2. Les Modules npm

Node.js contient plusieurs modules qui ont différents fonctionnements mais leur but est de faciliter l'utilisation de Node.js.

- `express` : La création de l'application et de ses comportements.
- `https` : Créer le serveur HTTP sécurisé.
- `socket.io` : Pouvoir interagir en temps réel avec les clients et le serveur.
- `easyrtc` : Pouvoir utiliser le Framework EasyRTC, et communication media peer-to-peer.
- `serve-static` : permet de créer une fonction de middleware pour servir des fichiers depuis un répertoire racine donné.

Notons qu'il existe beaucoup d'autres modules npm pour Node.js qu'on aurait pu utiliser.

La figure (fig. VI-2) ci-dessous est un extrait du fichier JavaScript du serveur qui montrer les différents modules utilisés dans notre application.

```

1 // Load required modules
2 var http    = require("https");           // http server core module
3 var express = require("express");        // web framework external module
4 var serveStatic = require('serve-static'); // serve static files
5 var socketIo = require("socket.io");     // web socket external module
6 var easyrtc = require("../");

```

Figure IV- 2: Liste des Modules utilisés.

3.1.3. Socket.io

▪ Définition

Socket.io est à sa version v1.3.5, considérée comme le plus rapide et le plus fiable moteur en temps réel, puisqu'elle permet une communication bidirectionnelle basée sur les événements. Elle marche pratiquement sur toutes les plateformes, navigateurs ou appareils où la fiabilité et la vitesse sont importantes. [26]

▪ Exemple d'utilisation

- Analyses en temps réel : Fournir des données aux clients responsables des calculs en temps réel, des graphiques ou des logs.
- Le streaming binaire : Depuis la version 1.0, il est possible d'envoyer n'importe quel type de données média en avant ou en arrière : vidéos, images, audio.
- Messagerie instantanée ou chat
- La collaboration de document : Autoriser les utilisateurs à modifier simultanément un document et voir les changements des autres utilisateurs.

▪ Principe de fonctionnement

La communication entre utilisateur et serveur, se fait très facilement avec les socket.io, en utilisant les événements à chaque nouvelle émission de messages. Comme dans le cas suivant, se lance l'événement 'connection' quand l'interface client (navigateur) se connecte au serveur via les socket.io, dedans se trouve les autres événements qui s'exécutent si un client lui émet un message.

```

44 var socketServer = socketIo.listen(webServer, {"connection":1});
45
46 easyrtc.setOption("connection", "debug");

```

Figure IV- 3: Extrait du code d'utilisation de socket.io.

3.1.4. EasyRTC

▪ Définition

EasyRTC est un outil WebRTC complet open source. C'est une trousse appropriée pour la construction hautement sécurisée d'applications d'entreprise avec WebRTC. Il s'agit d'un paquet d'applications web, des extraits de code, des bibliothèques de composants client et serveur méticuleusement écrites et documentées. Avec EasyRTC, les développeurs Web peuvent se mettre au diapason et obtenir des applications professionnelles qui utilisent WebRTC sur le marché. [27]

3.2. Le Serveur ROS

3.2.1. ROS (Robot Operating System)

▪ Définition

Comme son nom l'indique, ROS (Robot Operating System) est un système d'exploitation pour robots. De même que les systèmes d'exploitation pour PC, serveurs ou appareils autonomes, ROS est un système d'exploitation complet pour la robotique de service.

ROS est un méta système d'exploitation, quelque chose entre le système d'exploitation et le middleware.

Il fournit des services proches d'un système d'exploitation (abstraction du matériel, gestion de la concurrence, des processus...) mais aussi des fonctionnalités de haut niveau (appels asynchrones, appels synchrones, base de données centralisée de données, système de paramétrage du robot...). [28]

▪ L'organisation générale de ROS

La philosophie de ROS se résume dans les 5 grands principes suivants :

- **Peer to Peer:** Les programmes individuels se concentrent sur une API définie (messages ROS, services, etc.).
- **Distribué:** Les programmes peuvent fonctionner sur plusieurs ordinateurs et communiquer sur le réseau
- **Multi langages:** Les modules ROS peuvent être écrits dans n'importe quel langage pour lequel une bibliothèque cliente est disponible (C ++, python, MATLAB, Java, etc.)
- **Léger:** Les bibliothèques autonomes sont entourées d'une fine couche ROS

- **Gratuit et open source** : la plupart des logiciels ROS sont ouverts et gratuits.

3.2.2. Transformation des données

La figure suivante représente la transformation des données entre les nœuds du robot :

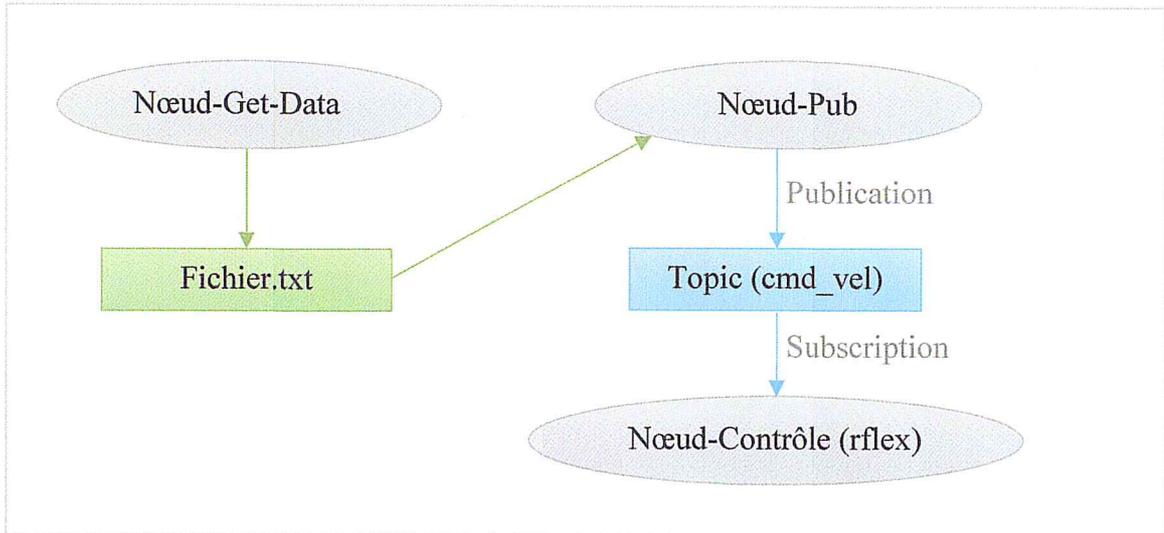


Figure IV- 4: transformation des données entre les nœuds.

Le Nœud Get-Data c'est le responsable de la récupération des données qui sont envoyées par l'utilisateur, et sont stockées dans « fichier.txt ».

Dans « fichier.txt » les données (la vitesse angulaire et la vitesse linéaire) sont modifiées à chaque fois que le nœud Get-Data reçoit une nouvelle donnée.

Le Nœud pub récupère les données qui sont stockées dans le fichier.txt, puis les publie dans le topic (cmd_vel).

Le nœud de contrôle (rflex) fait la subscription des données, et à partir de ces données il gère le robot.

- **Topic** : est un mode de communication asynchrone permettant une communication many-to-many. Le **service** en revanche répond à une autre nécessité, celle d'une communication synchrone entre deux nœuds.

3.2.3. Web Socket

- **Définition**

Une WebSocket est une spécification d'un protocole permettant une communication bidirectionnelle, et full duplex sur une seule socket TCP entre le client et le serveur.

Initialement développé pour HTML 5. WebSocket a été normalisé par l'IETF et le W3C. Tous les navigateurs récents implémentent et supportent les WebSockets.

Ce protocole permet notamment d'implémenter facilement et de manière standard l'envoi des données en mode Push à l'initiative du serveur. [28]

- La figure suivante présente l'utilisation de Websocket côté client :

```
22 // Connect to Web Socket
23 ws = new WebSocket("ws://192.168.43.229:9002/");
24
25 // Set event handlers.
26 ws.onopen = function() {
27     output("onopen");
28 };
29
30 ws.onmessage = function(e) {
31     // e.data contains received string.
32     output("onmessage: " + e.data);
33 };
34
35 ws.onclose = function() {
36     output("onclose");
37 };
38 ws.onerror = function(e) {
39     output("onerror");
40     console.log(e)
41 };
```

Figure IV- 5: Extrait du code d'utilisation de Websocket coté Client.

- La figure suivante présente l'utilisation de WebSocket côté serveur:

```

1  from websocket_server import WebSocketServer
2  PORT=9002
3  url1='V_lineaire.txt'
4  url2='V_angulaire.txt'
5  def UpDataInText(url,txt):
6      with open('modified.mpd', 'w') as mpd:
7          txt=str(txt)
8          data = mpd.write(txt)
9      with open(url,'w') as mpd:
10         data=mpd.write(txt)
11 def getDataWithTxt(url):
12     with open(url,'r') as btx:
13         dbtx = btx.read()
14         data=float(dbtx)
15         print("file ",url,"data=",data)
16 def Update_V_L(val):
17     val=float(val)
18     UpDataInText('V_lineaire.txt',val)
19 def Update_V_A(val):
20     val=float(val)
21     UpDataInText('V_angulaire.txt',val)
22 #print ('listen for IP: 127.0.0.1 and Port: ',PORT)
23 # Called for every client connecting (after handshake)
24 def new_client(client, server):
25     print("New client connected and was given id %d" % client['id'])
26     #server.send_message_to_all("Hey all, a new client has joined us")
27 # Called for every client disconnecting
28 def client_left(client, server):
29     print["Client(%d) disconnected" % client['id']]
30 # Called when a client sends a message
31 def message_received(client, server, message):
32     server = WebSocketServer(PORT)
33     server.set_fn_new_client(new_client)
34     server.set_fn_client_left(client_left)
35     server.set_fn_message_received(message_received)
36     server.run_forever()

```

Figure IV- 6: Extrait du code d'utilisation de WebSocket coté serveur ROS.

4. Présentation de l'application

4.1. Interface principale du robot

Une fois connecté au robot, l'utilisateur dispose de plusieurs fenêtres pour le contrôle de celui-ci :

- En rouge (Vue de tête) : Capteur de caméra locale (Robot).
- En vert (Vue d'utilisateur) : Capteur de caméra d'utilisateur.

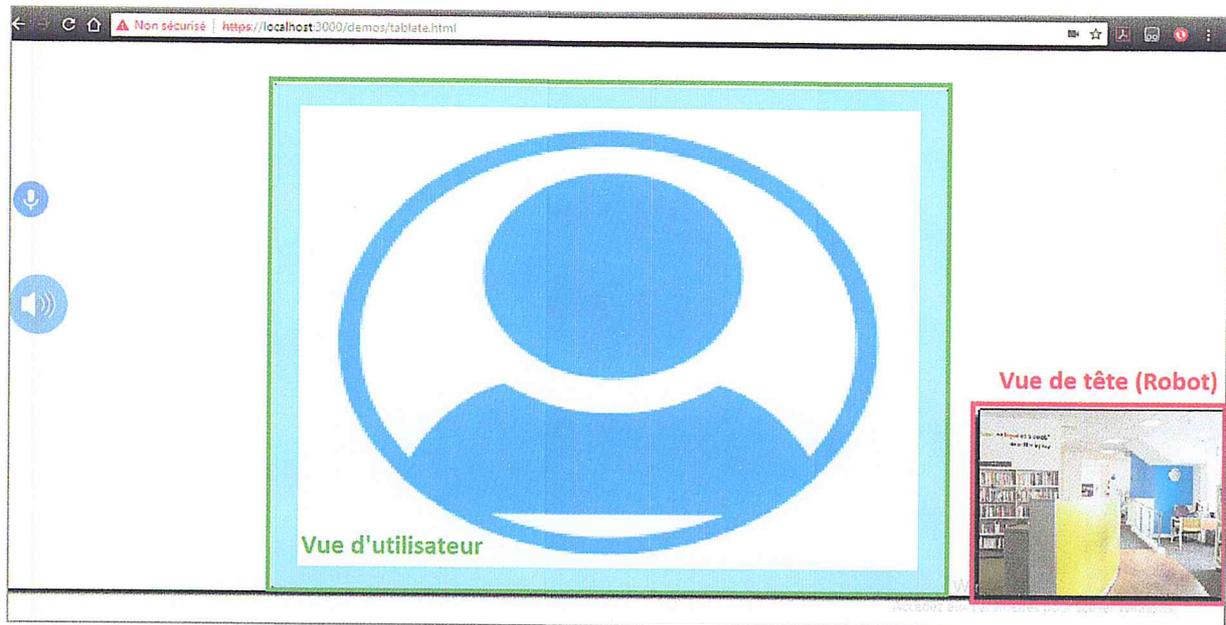


Figure IV- 7: Interface principale du robot.

4.2. Interface d'authentification

Au lancement de notre application, un formulaire s'affiche à l'écran, il demande d'introduire le login et le mot de passe d'authentification pour accéder au menu principal.

L'administrateur saisit son login et mot de passe, ensuite il doit cliquer sur le bouton connexion (figure IV-10).

Le système vérifie dans la base de données les informations saisies. Dans le cas où les informations saisies sont correctes, l'administrateur accède à l'application, dans le cas contraire un message d'erreur sera affiché (figure IV-8).

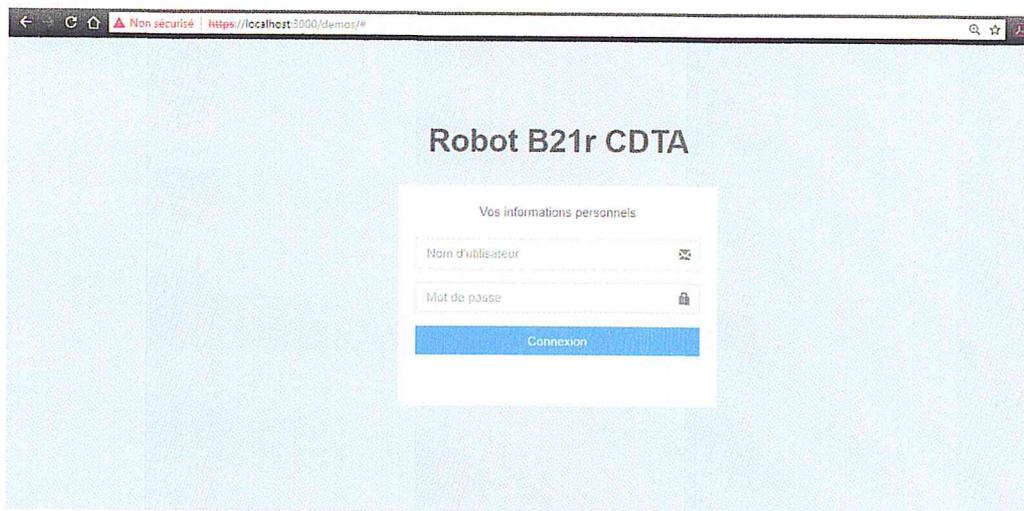


Figure IV- 8: Interface « d'authentification ».

La figure suivante représente l'interface de message d'erreur d'authentification :

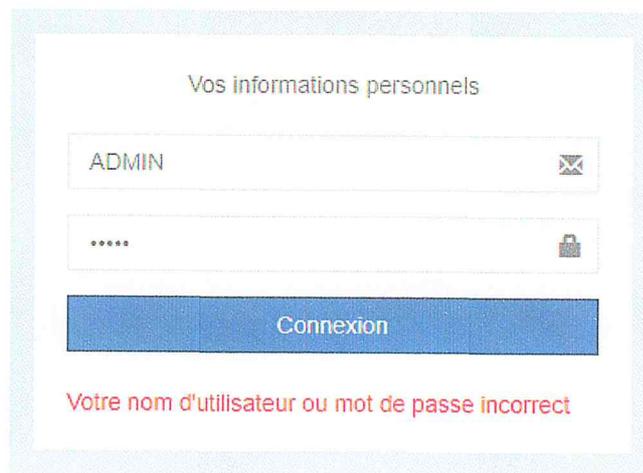


Figure IV- 9: Interface « Message d'erreur d'authentification ».

4.3. Interface principale « Accueil »

Cette interface comporte le menu principal où l'utilisateur pourra sélectionner la tâche à effectuer. (Figure IV-10).

Ce menu contient cinq (05) boutons qui sont:

- **Accueil:** Pour la présentation du robot.
- **Manipulation du robot :** Pour se connecter et contrôler le robot.
- **Liste des utilisateurs:** Pour afficher la liste des utilisateurs, modifier, ajouter, supprimer et activer/désactiver les utilisateurs ; ce bouton ne s'affiche qu'à l'administrateur.
- **Réglages:** Modifier les informations personnelles.
- **Déconnecter:** pour déconnecter l'application et revenir à la page d'authentification.



Figure IV- 10: Interface « Principale ».

4.4. Interface de Manipulation du robot

Une fois l'utilisateur connecté, il dispose de différentes fenêtres dans l'interface de l'application (figure IV-11).

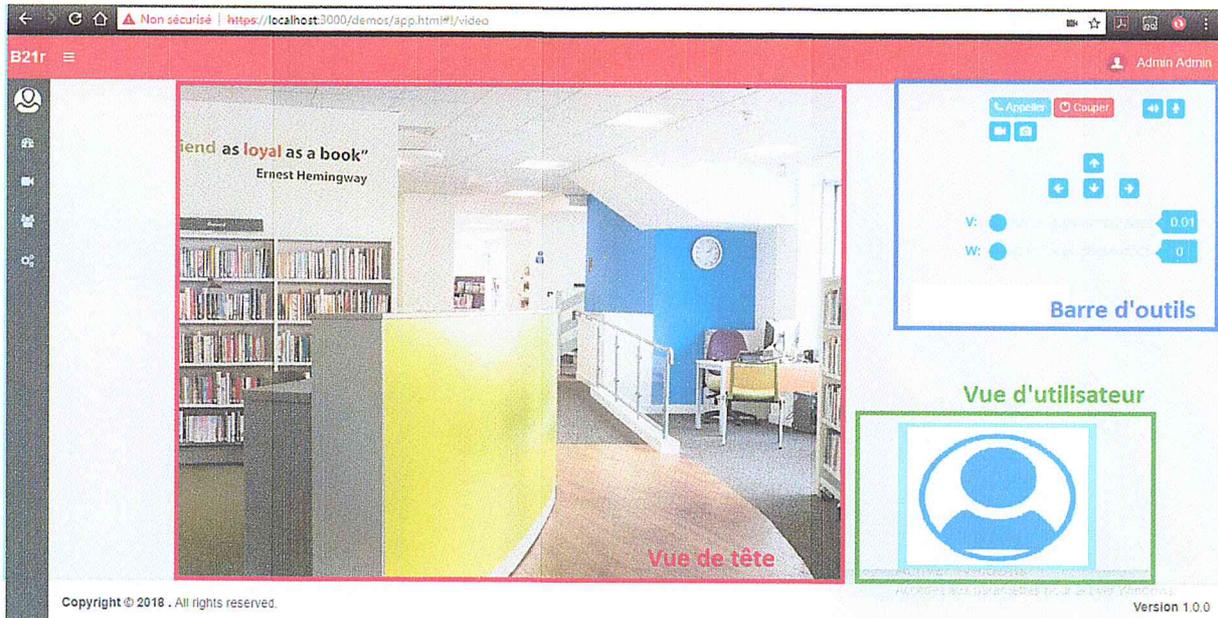


Figure IV- 11: Interface de Manipulation du robot .

- En rouge (Vue de tête): Capture de la caméra du robot.
- En vert (Vue d'utilisateur): Capture de la caméra locale (Utilisateur).
- En bleu (Barre d'outils): Voir la figure IV-12.

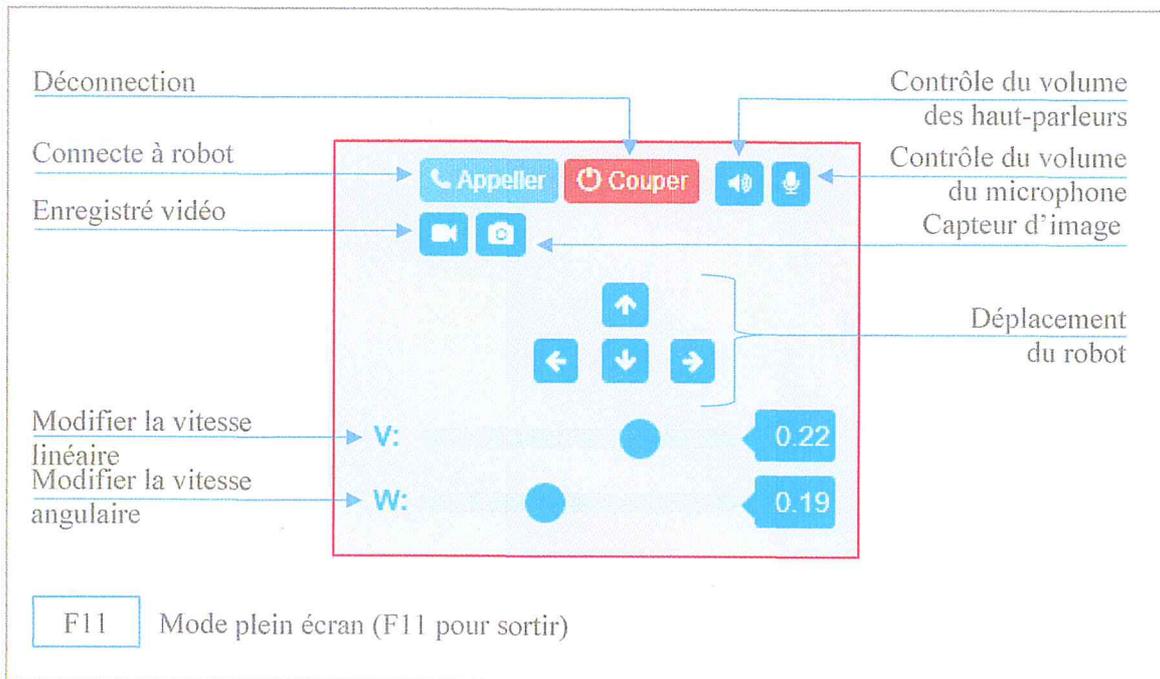


Figure IV- 12: Présentation de barre d'outils.

4.5. Interface « Déconnexion »

Ce bouton permet de quitter l'application et revenir à la page d'authentification.

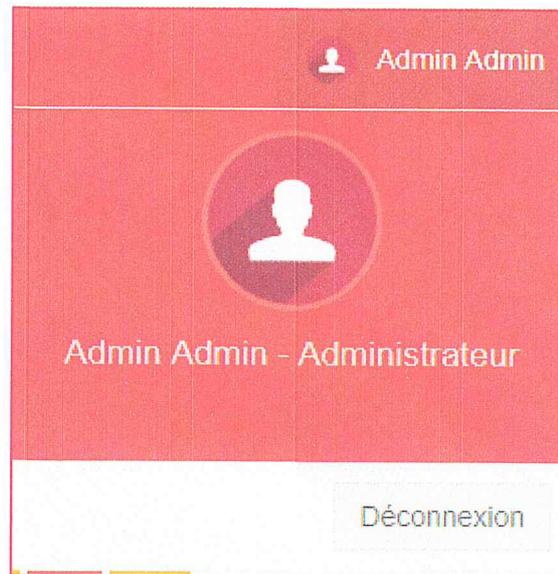


Figure IV- 13: Interface « Déconnexion ».

4.6. Les différents profils (Administrateur/Utilisateur)

4.6.1. Menu de l'utilisateur



Figure IV- 14: Interface « Menu de l'utilisateur ».

- **Interface « Réglages - Informations personnelles »**

Changer vos informations

Nom test

Prenom test

Annuler Valider

Figure IV- 15: Interface « Modifier les Information personnel ».

- **Interface « Réglages – Mot de passe »**

Changer votre mot de passe

Ancien mot de passe Ancien mot de passe

Nouveau mot de passe Nouveau mot de passe

Confirme mot de passe Confirme mot de passe

Annuler Valider

Figure IV- 16: Interface « Modifie le mot de passe ».

4.6.2. Menu de l'administrateur



Figure IV- 17: Interface « Menu de l'administrateur ».

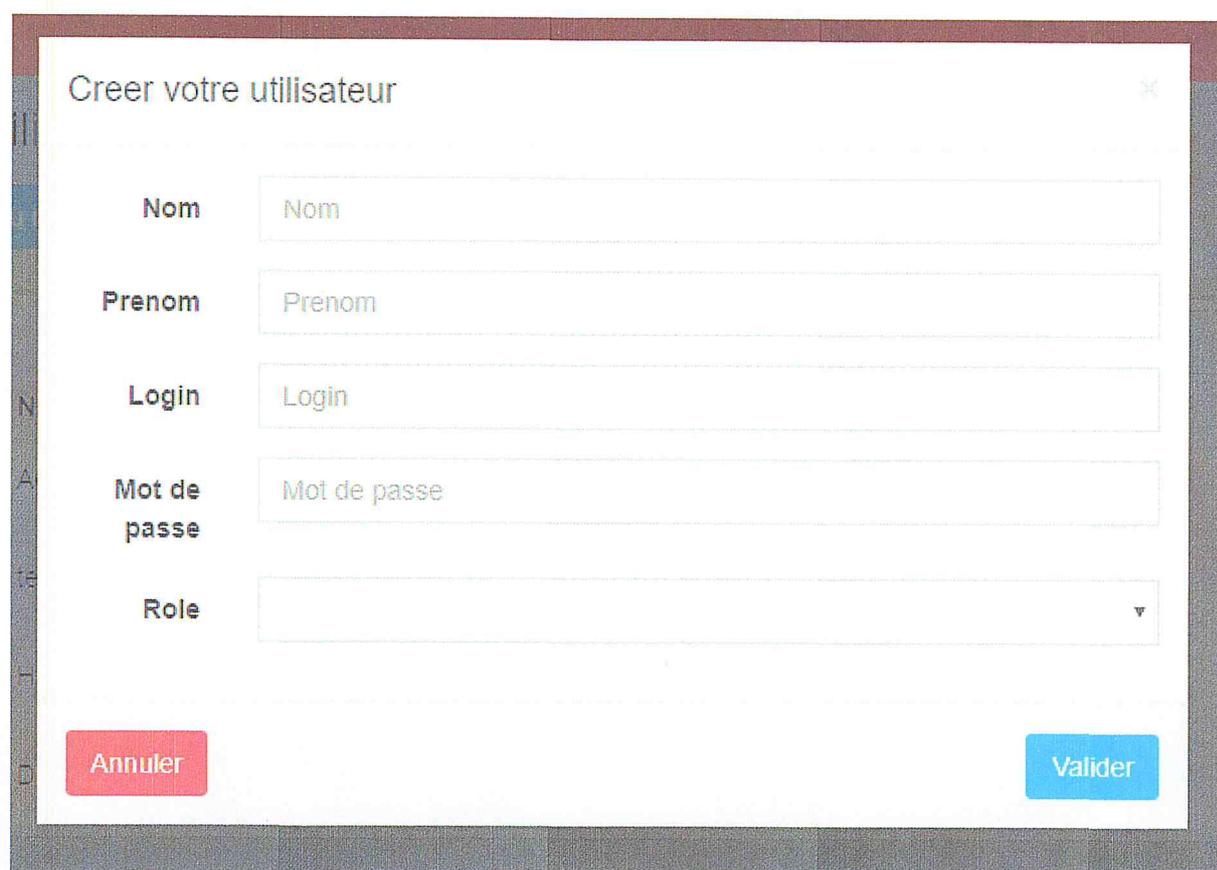
▪ Interface « Lister les utilisateurs » :

Login	Nom	Prenom	Role	Etat	Action
admin	Admin	Admin	Administrateur	Activé	[Edit] [Delete] [Search]
test	test	test	Utilisateur	Activé	[Edit] [Delete] [Search]
aaa	Handi	Adel	Utilisateur	Desactivé	[Edit] [Delete] [Search]
bbb	Dlaber	Salah	Administrateur	Desactivé	[Edit] [Delete] [Search]

Figure IV- 18: Interface « Lister les utilisateurs »

- **Interface d'inscription d'un nouvel utilisateur :**

L'inscription à notre application web nécessite un nom, prénom et un mot de passe. Une fois le bouton « valider » cliqué, le processus d'inscription vérifie si l'identifiant n'a pas déjà été utilisé, si oui l'utilisateur sera ajouté à la base de données, et sera automatiquement redirigé vers la page d'accueil, sinon un message d'erreur s'affichera en précisant le problème qui a fait que l'inscription a échoué.



The image shows a web form titled "Créer votre utilisateur" (Create your user) with a close button (X) in the top right corner. The form contains five input fields: "Nom" (Name), "Prenom" (First name), "Login", "Mot de passe" (Password), and "Role" (Role). The "Role" field is a dropdown menu. At the bottom left, there is a red "Annuler" (Cancel) button, and at the bottom right, there is a blue "Valider" (Validate) button.

Figure IV- 19: Interface d'inscription d'un nouvel utilisateur.

5. Conclusion

Nous avons présenté dans ce chapitre les détails techniques liés à la mise en œuvre de notre système. Nous avons commencé par présenter le robot mobile B21r, après on a cité les différents langages utilisés, ainsi que les technologies logicielles utilisées. Nous avons ainsi utilisé Node.js et ROS pour la partie serveur, les websockets pour la communication entre les navigateurs et les serveurs, et enfin EasyRTC pour la communication peer-to-peer entre les navigateurs.

La fin du chapitre a été consacrée à la présentation des différentes interfaces et fonctionnalités de l'application.

Conclusion générale
& Perspective

Conclusion générale et perspectives

Le travail de notre projet de fin d'étude que nous avons effectué a pour thème le développement et l'implémentation d'une commande à distance pour un robot de surveillance. Ce travail consiste à créer un système de surveillance à base d'un robot mobile existant. Le robot a été agrémenté d'un système embarqué disposant du wifi, ainsi que d'une webcam mobile.

L'application web que nous avons réalisée utilise les technologies web temps réel, qui sont utilisées dans les navigateurs web. Ces technologies incluent le protocole Websockets qui permet la communication bidirectionnelle entre le serveur et le client. Il est soutenu par tous les navigateurs web modernes. Nous avons aussi utilisé EasyRTC pour gérer ensuite les chambres de connexion et établir des appels entre les pairs. Nous avons utilisé le ROS pour contrôler le robot de surveillance.

Pour la conception de notre application, nous avons utilisé le langage UML qui permet de prendre en compte les spécifications de l'application web.

Pour la réalisation, nous avons utilisé le langage de programmation JavaScript. L'utilisation de ce langage a permis d'avoir un code léger que ce soit du côté client et du côté serveur, assurant ainsi de hautes performances pour notre application web.

Bien que notre application ne soit qu'à sa première version, elle est néanmoins totalement fonctionnelle et offre un certain nombre de fonctionnalités intéressantes telles que :

- La possibilité de faire un appel vidéo par son et image en temps réel.
- La possibilité de manipuler un robot via notre application.
- La possibilité de contrôler la vitesse souhaitée.

Dans la prochaine version de notre application web, nous envisageons de rajouter les fonctionnalités suivantes :

- Mettre en place un dispositif de commande vocal.
- Accès au site par empreinte digitale.
- Augmenter la qualité de la vidéo par augmentation de la qualité de la camera et la bande passante de connexion entre l'application web et le robot.
- Equiper le robot avec des détecteurs afin de détecter la température et envoyer toutes les informations du lieu de travail.

Bibliographie

- [16] Zig Bee Specification FAQ". Zigbee Alliance. Retrieved 14 June 2013. ZigBee Wireless Networking", Drew Gislason (via EETimes).
- [17] <http://www.wi6labs.com/2016/06/21/quelle-technologie-radio-pour-les-objets-connectes-deuxieme-partie/> consulté : mai 2018.
- [18] Pierre-Alain Muller – Modélisation objet avec UML, Eyrolles 1997.
- [19] Pascal Roques, UML 2: modéliser une application web, Les Cahiers du programmeur, Eyrolles, 2008.
- [20] Pascal Roques, « UML2 par la pratique », 5eme édition, 2006.
- [21] Booch G, Rumbaugh J, Jacobson I. « Orienté objet avec UML », Editions EYROLLES, 2000.
- [22] Luc Van Lancker, Des CSS au DHTML: JavaScript appliqué aux feuilles de style, Ediciones ENI, 2008.
- [23] <https://www.python.org/>.
- [24] Mathieu Nebra, <http://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/node-js-mais-a-quoi-ca-sert,2018>.
- [25] Mathieu Nebra, <http://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/les-modules-node-js-et-npm,2018>.
- [26] <https://socket.io/>.
- [27] www.ros.org/.
- [28] <http://easyrtc.com/>.
- [29] Andrii Sergiienko , WebRTC Cookbook, Packt Publishing Ltd, 2015.

