

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITÉ FERHAT ABBAS - SETIF1

FACULTÉ DE TECHNOLOGIE

THÈSE

Présentée au Département d'Electronique

Pour l'obtention du diplôme de

DOCTORAT

Domaine : Sciences et Technologie

Filière : Télécommunications

**Option : Réseaux et
Télécommunications**

Par

TAMALI Abderrahmane

THÈME

**Contribution au développement d'un cluster de
micro-cobots pour les sites hostiles**

Soutenue le 10/06/2024 devant le Jury:

FERHAT HAMIDA Abdelhak	Professeur	Univ. Ferhat Abbas Sétif 1	Président
AMARDJIA Nourredine	Professeur	Univ. Ferhat Abbas Sétif 1	Directeur de thèse
TAMALI Mohammed	Professeur	Univ. Tahri Mohammed Bechar	Co-Directeur
BOUGUEZEL Saad	Professeur	Univ. Ferhat Abbas Sétif 1	Examineur
BEKKOUCHE Tewfik	M.C.A.	Univ. Mohamed El Bachir El Ibrahimi de BBA	Examineur
KHOUNI Habib	M.C.A.	Univ. Ferhat Abbas Sétif 1	Examineur

Dédicaces

Avec l'aide de Dieu le tout puissant et miséricordieux, nous dédions ce modeste travail à tous ceux que nous aimons, chérissons et respectons.

En ce moment des plus rudes et pénibles de l'histoire de l'humanité, nous n'hésitons à clamer le droit légitime de nos frères Palestiniens sous l'oppression et l'occupation, à une vie sous un état libre et prospère. A tous les Palestiniens et à toute personne libre, nous dédions ce travail.

A mes parents, surtout à la mémoire de ma chère défunte mère, nous nous tenons debout pour tout bon souffle et souhait laissé en faveur de ma réussite.

Remerciements

Je tiens à exprimer mes sincères remerciements à mon encadreur, le Professeur AMARDJIA Nourredine, pour sa patience, son cœur généreux, sa disponibilité et son dévouement sans faille. À mon cher père TAMALI Mohammed, qui a été un co-encadreur dévoué, je vous exprime ma plus profonde gratitude et mon amour éternel. Votre soutien indéfectible a été une source d'inspiration tout au long de ce parcours. Mes respects vont également à tous les membres du jury pour avoir accepté d'évaluer ce travail. Je suis reconnaissant envers mes chers Professeurs pour leur soutien et leurs efforts visant à valoriser mes travaux de thèse. Un grand merci à toute la communauté universitaire de l'Université Ferhat Abbas, qui m'a accueilli et soutenu tout au long de ce parcours académique. Mes remerciements vont également aux membres du CFD pour leur hospitalité et leur soutien pendant mon doctorat.

Enfin, je souhaite exprimer ma gratitude à tous ceux qui m'ont enseigné, conseillé et encouragé au cours de mes études de graduation et tout au long de ma première année de doctorat. Votre influence a été déterminante dans la réalisation de ce travail.

Table de matières

Liste des abréviations.....	i
Liste de figures.....	ii
Liste de tableaux.....	v
Liste des algorithmes.....	v
Problématique.....	1
Introduction générale.....	3

Chapitre I : Etude théorique

I.1. Introduction.....	7
I.2. Développement des robots.....	8
I.3. Les critères relatifs à la morphologie et à la mobilité.....	13
I.4. La robotique et les télécommunications.....	16
I.5. La Cobotique.....	17
I.5.1. Naissance de la cobotique.....	19
I.5.2. Classification en fonction du rôle de l'opérateur.....	22
I.5.3. Autonomie des cobots.....	24
I.5.4. Cobotique et ses applications.....	25
I.6. Collaboration multirobots.....	25
I.6.1. Classement par taille de l'équipe.....	26
I.6.2. Classification par morphologie et capacités des robots.....	27
I.6.3. Classification par niveau de coordination entre les systèmes multirobots.....	28
I.6.4. Architecture logicielle pour les systèmes multirobots.....	30
I.7. Conclusion.....	33

Chapitre II : Connectivité des systèmes multirobot et réseaux de capteurs sans-fils (WSN)

II.1. Introduction.....	35
II.2. Problèmes de la connectivité.....	36
II.3. Quantification et caractérisation de la fluctuation de la qualité des liens.....	38
II.4. Caractéristiques des liens mobiles.....	40
II.4.1. Caractéristiques spatiales.....	40
II.5. Prédiction de la qualité des liens de connectivité entre les WSN.....	42
II.5.1. Travaux connexes pour la prédiction de la connectivité.....	42
II.5.2. GMLA pour la prédiction de la qualité des liaisons dans les WSNs.....	46
II.6. Analyse des technologies de transmission sans fil (systèmes multirobots mobiles).....	51
II.6.1. Technologies de transmission pour robots mobiles.....	53

II.6.2. Méthodologie Expérimentale.....	56
II.6.3. Résultats et Discussion.....	59
II.7. Conclusion.....	64

Chapitre III : Sécurité des réseaux IoT pour multirobots

III.1. Introduction.....	66
III.2. IoT et la sécurisation des réseaux.....	68
III.3. Sécuriser les robots mobiles collaboratifs avec l'IoT.....	68
III.4. Contrôle continu.....	68
III.5. Détection et réponse aux menaces.....	69
III.6. Cryptage des données.....	70
III.7. Gestion à distance et mises à jour.....	70
III.8. Authentification et contrôle d'accès.....	71
III.9. Conclusion.....	73

Chapitre IV : Allocation de tâches pour multirobot (MRTA)

IV.1. Introduction.....	74
IV.2. Allocation de tâche selon l'architecture du réseau multi-nœud.....	75
IV.2.1. Allocation centralisée des tâches multi-nœuds d'agents.....	75
IV.2.2. Allocation de tâches multi-nœuds d'agents décentralisée.....	76
IV.2.3. Allocation de tâches multi-nœuds d'agents hybride.....	77
IV.3. Travaux connexes pour des systèmes d'allocation de tâches multi-nœuds d'agents.....	77
IV.3.1. Méta heuristique (Meta Heuristics).....	77
IV.3.2. Intelligence d'une configuration en essaim.....	79
IV.3.3. Approches basées sur le marché (Market-Based).....	81
IV.4. Algorithmes pour l'attribution de tâches multi-capteurs.....	83
IV.4.1. Algorithme des abeilles distribuées (Distributed Bees Algorithm).....	83
IV.4.2. Algorithme des abeilles distribuées hétérogènes.....	85
IV.4.3. Système d'abeille.....	87
IV.4.4. Optimisation des colonies de fourmis.....	87
IV.4.5. Algorithme basé sur le marché (l'enchère).....	88
IV.4.6. Fisher Market Clearing Task Allocation (FMC_TA).....	90
IV.4.7. Algorithme glouton.....	91
IV.4.8. Algorithme génétique.....	94
IV.4.9. Recuit simulé (Simulated Annealing).....	95
IV.5. Discussion.....	96
IV.6. Système d'exploitation en temps réel dans la MRTA	97
IV.6.1 Avantages d'un RTOS dans un système MRTA	97

IV.6.2 Intégration d'un RTOS dans MRTA	98
IV.7. Conclusion.....	98
<u>Chapitre V : Conception d'AlMustaksheef3D et du VLiDAR3D</u>	
V.1. Introduction.....	100
V.2. Présentation du vLiDAR3D.....	100
V.3. Modèle du microrobot AlMustaksheef3D.....	104
V.4. Conclusion.....	111
<u>Chapitre VI : Etude de cas</u>	
VI.1. Introduction.....	113
VI.2. Le Framework ROS.....	113
VI.2.1. Utilisation de ROS pour les systèmes multirobots.....	115
VI.3. Préparation de la scène.....	116
VI.4. Processus de collaboration et d'affectation de taches utilisé.....	119
VI.5. Énoncé du problème.....	120
VI.6. Algorithme d'optimisation.....	124
VI.7. Collaboration par algorithme Greedy pour le MRTA distribué.....	129
VI.8. Spécifications adoptées du système d'exploitation et du poste de travail.....	132
VI.9. Scenarios et résultats.....	132
VI.10. Discussion et perspectives.....	142
VI.11. Conclusion.....	145
Conclusion générale.....	146
Références.....	149
Annexes.....	170

Liste des abréviations

ABC : Artificial Bee Colony	LTE : Long Terme Evolution
ACK : Acknowledgment	MAC : Message Authentication Code
ACL : Access Control List	MANET : Mobile Ad hoc Network
ACO : Ant Colony Optimization	MAS : Multi agent System
AG : Genetic Algorithm	MaSE : Multi agent System Engineering
AG/GA : Genetic Algorithm	MCU : Microcontroller Unit
AP : Access Point	MEMs : Micro Electro-Mechanical Systems
API : Application Programming Interface	MFA : Multi-factor Authentication
ARR : Acquired Reception Ratio	MQTT : Message Queuing Telemetry Transport
BA : Bee Algorithm	MRTA : Multi-Robot Task Allocation
BCO : Bee Colony Optimization	MTCP : Multipath TCP
BD : Birth-Death	NAT : Network Address Translation
BEEI : Bulletin of Electrical Engineering and Informatics	NLOS : Non-Line of Sight
BS : Bee System	NP : Nondeterministic Polynomial time
CFLA : Coalition Formation with Look Ahead	OAP : Optimal Assignment Problem
CFSTP : Coalition Formation with Spacial and Temporal constrains Problem	OBD : Oriented Birth-Death
CNP : Contrat Net Protocol	OS : Operating System
CoAP : Constrained Application Protocol	OTA : Over The Air
CPU : Central Processing Unit	PCB : Printed Circuit Board
DBA : Distrubed Bee Algorithl	PCD : Point Cloud
DCOP : Distributed Constraint Optimization Problem	PDR : Packet Delivery Ratio
DDL : Degre De Liberte	PKI : Public Key Infrastructure
DOF : Degree Of Freedom	PRR : Packet Reception Ratio
DoM :Degré de Manœuvrabilité	PSR : Packet Sending Ratio
DoS : Degree of Steerability	RBAC : Role Based Access Control
DSN : Delivery Status Notification	RF : Radio Frequency
DTLS : Datagram Transport Layer Security	RGB : Red-Green-Blue
FMC-TA : Fisher Market Cleaning - Task Allocation	ROI : Region Of Interest
Gen AI : Genetic AI	ROS : Robot Operating System
GMLA : Genetic Machine Learning Approach	RSSI : Received Signal Strength Indicator
GPS : Global Positioning System	RTOS : Real Time Operating System
GPT : Generative Pre-training Transforms	RUR : Rossum’s Universal Robot
GPU :Graphics Processing Unit	Rviz : ROS visualisation
HBMO : Honey-Bee Mating Optimization	SA : Simulated Annealing
HDBA : Heterogeneous Distributed Bee Algorithm	SCADA : Supervisory Control And Data Acquisition
IA : Intelligence Artificielle	SDF : Simulation Description Format
IEEE : Institute of Electrical & Electronics Engineers	SIEM : Security Information & Event Management
IMU : Inertial Measurement Unit	SLAM : Simultaneous Localization And Mapping
INRS : Institut National de Recherche et de Sécurité	SNR : Signal to Noise Ratio
IoT : Internet of Things	TCP/IP : Transport Control Protocol/Internet Protocol
IPI : Inter Packet Interval	TDOA : Time Difference Of Arrival
IPv6 : Internet Protocol version 6	TLS : Transport Layer Security
ISO : International Standard Organization	UAV : Unmanned Air Vehicle
LBR : Leichtbauroboter	URDF : Universal Robotic Description Format
LiDAR : Light Detection And Ranging	UR5 : Universal Robot 5
LLM : Large Language Model	VLiDAR3D : Vertical Lidar 3D
LoRa : Long Range	VPN : virtual private network
LOS : Line of Sight	WiFi : Wireless Fidelity
LQE : Link Quality Estimator	WLAN : Wireless LAN
LQI : Link Quality Indicator	WSN : Wireless Sensor Network
LQP : Link Quality Prediction	2D/3D : Two Dimentional/ Three Dimentional
	3/4/5G : 3 rd /4 th /5 th Generation

Liste de figures

Fig. : Véhicule ROVER M2020, Crédit NASA/JPL-Caltech. Véhicule conçu pour des missions spatiale sur MARS.....	2
Fig. I.1 : le renard de Ducrocq.....	8
Fig. I.2 : Le robot Unimate de Unimation-General Motors.....	8
Fig. I.3 : <i>Linokhod 1</i> , engins spatiale Russe, envoyés sur la lune.....	9
Fig. I.4 : des robots (ANYmal, ANYMAL C d'ANYBOTICS, TESLA Optimus & Humanoïde.....)	9
Fig. I.5 : Robots de dimension NANO.....	10
Fig. I.6 : Structure de préparation d'un LLM.....	12
Fig. I.7 : Génoïde Sophia de Hanson Robotics.....	12
Fig. I.8: MRTA with LLM Based control.....	12
Fig. I.9 : Sortes de microrobots.....	13
Fig. I.10 : Un robot éliminant les obstacles dans les égouts	14
Fig. I.11 : Robot à 4 roues.....	14
Fig. I.12 : Robot à chenilles.....	14
Fig. I.13 : Robot Multipède.....	15
Fig. I.14 : Robot à mobilité hybride.....	15
Fig. I.15: Les petits robots <i>Kilobots</i>	15
Fig. I.16: Structure générale d'un robot à 2 DDL (Rotation+Translation).....	16
Fig. I.17 : Mise en réseau IoT du cluster de microrobots.....	17
Fig. I.18 : Classification des robots selon l'Institut National de Recherche et de Sécurité.....	18
Fig. I.19 : Schéma général d'un système cobotique.....	19
Fig. I.20 : Un déchargeur de porte au sol développé par l'Université Northwestern est testé dans une usine GM.....	20
Fig. I.21 : L'UR5 et LBR iiwa pour (a) et (b) respectivement.....	21
Fig. I.22 : Coût par unité de production en fonction du volume de production annuel pour la cobotique.....	22
Fig. I.23 : Des exemples de réseaux provinciaux et cosmopolites sont présentés.....	29
Fig. I.24 : Séisme de Boumerdes.....	33
Fig. I.25 : Les ruines de Casbah.....	33
Fig. I.26 : Lieu radioactif à Reggane.....	33
Fig. II.1 : Caractérisation de la fluctuation de la qualité des liaisons dans les environnements intérieurs et extérieurs.....	41
Fig. II.2 : Le SNR en fonction de la situation d'approche et de départ au fil du temps.....	46
Fig. II.3 : Automate orienté naissance mort (O-BD).....	46
Fig. II.4 : Schéma du système de classification.....	48
Fig. II.5 : Le chromosome en tant que matrice et les lignes comme gènes.....	49
Fig. II.6 : L'approche d'apprentissage automatique génétique pour la prédiction de la qualité des liens.....	50
Fig. II.7 : les modules de transmissions utilisés.....	57
Fig. II.8 : Illustration de la configuration expérimentale.....	58
Fig. II.9 : RSSI en fonction de la distance, la technologie ZigBee.....	59
Fig. II.10 : SNR en fonction de la distance, la technologie ZigBee.....	59
Fig. II.11 : Débit en fonction de la distance, la technologie ZigBee.....	60
Fig. II.12 : RSSI en fonction de la distance, la technologie Bluetooth.....	60

Fig. II.13 : SNR en fonction de la distance, la technologie Bluetooth.....	60
Fig. II.14 : Débit en fonction de la distance, la technologie Bluetooth.....	61
Fig. II.15 : RSSI en fonction de la distance, la technologie LoRa.....	61
Fig. II.16 : SNR en fonction de la distance, la technologie LoRa.....	61
Fig. II.17 : Débit en fonction de la distance, la technologie LoRa.....	62
Fig. II.18 : RSSI en fonction de la distance, la technologie Wifi.....	62
Fig. II.19 : SNR en fonction de la distance, la technologie Wifi.....	62
Fig. II.20 : Débit en fonction de la distance, la technologie Wifi.....	63
Fig. III.1 : Configuration de base d'un circuit IoT (bout-en-bout).....	66
Fig. III.2 : Un système réseau IoT sécurisé pour la gestion des laboratoires.....	67
Fig. III.3 : Un schéma de déploiement représente un réseau IoT d'un laboratoire avec des objets.....	68
Fig. III.4 : Un prototype réalisé de processus de contrôle d'accès d'un réseau IoT avec quelques objets que le réseau renferme.....	68
Fig. IV.1 : Schéma d'allocation des tâches d'un système multi-capteur.....	75
Fig. IV.2 : Diagramme de flux de travail pour l'algorithme basé sur le marché.....	89
Fig. V.1 : Prototypes en voie de développement, <i>ALMustaksheef3D</i> , (a) Le microrobot. (b) le Lidar.....	100
Fig. V.2 : Lidar développé (vLiDAR3D) et degrés de liberté.....	101
Fig. V.3 : Deux configuration de LiDAR horizontal.....	102
Fig. V.4 : Planche de dessin technique du vLiDAR3D proposé.....	103
Fig. V.5 : Deux positions du RPLIDAR A1M8.....	103
Fig. V.6 : Représentation URDF 3D du robot.....	106
Fig. V.7 : Référentiel lié à la base du microrobot.....	107
Fig. V.8 : vLiDAR3D conçu, imprimé et développé avec ses composantes.....	107
Fig. V.9 : Topologie du microrobot à décrire en modèle XML-URDF.....	108
Fig. V.10 : Versions du microrobot <i>ALMustasheef3D</i> durant son développement.....	109
Fig. V.11 : Morphologie et composition générale du vLiDAR3D sans équipements.....	109
Fig. V.12 : Vue détail, principaux composants actifs.....	109
Fig. V.13 : Vue détail, MCU et IMU.....	110
Fig. V.14 : Vue détail, base fixe.....	110
Fig. V.15 : Vue de côté, détail de l'appareil.....	110
Fig. V.16 : Logiciels GPL Utilisés pour le développement.....	111
Fig. VI.1 : Le système de fichier de ROS.....	114
Fig. VI.2 : Outil éditeur de bâtiments de Gazebo.....	117
Fig. VI.3 : Outil éditeur de modèles de Gazebo.....	117
Fig. VI.4 : Modèle 3D d'un humain pour émuler un humain blessé.....	117
Fig. VI.5 : Caractéristique de décharge de courant nominal (0,65 A).....	118
Fig. VI.6 : <i>AL Mustaksheef3D</i> , développement d'un robot à roues.....	119
Fig. VI.7 : Bases et concepts de ROS.....	119
Fig. VI.8 : Architecture du cluster.....	120
Fig. VI.9 : N-gon d'une zone numérisée.....	121
Fig. VI.10 : Scénarios potentiels pour la posture du robot.....	123

Fig. VI.11 : Organigramme de notre approche.....	124
Fig. VI.12 : Algorithme génétique équivalent.....	129
Fig. VI.13 : JetBot pendant l'exploration du domaine.....	131
Fig. VI.14 : JetBot de Nvidia.....	132
Fig. VI.15 : Zone modèle de la simulation.....	133
Fig. VI.16 : Segmentation spatiale.....	133
Fig. VI.17 : Le modèle de robot utilisé (JetBot)	134
Fig. VI.18 : Robots lors de l'exécution d'une mission.....	134
Fig. VI.19 : Mises à jour du ROI des agents.....	134
Fig. VI.20 : Scènes avec emplacements robot-cibles.....	135
Fig. VI.21 : Temps mis par un seul robot pour atteindre les cibles (emplacement des cibles 1).....	136
Fig. VI.22 : Temps mis par un seul robot pour atteindre les cibles (emplacement des cibles 2).....	136
Fig. VI.23 : Temps mis par un seul robot pour atteindre les cibles (emplacement des cibles 3).....	136
Fig. VI.24 : Scènes avec emplacement robots-cibles.....	137
Fig. VI.25 : Temps mis par les deux robots pour atteindre les cibles (emplacement des cibles 1).....	137
Fig. VI.26 : Temps mis par les deux robots pour atteindre les cibles (emplacement des cibles 2).....	137
Fig. VI.27 : Temps mis par les deux robots pour atteindre les cibles (emplacement des cibles 3).....	138
Fig. V.28 : Scènes avec emplacement robots-cibles.....	138
Fig. VI.29 : Temps mis par les deux robots pour atteindre les cibles (emplacement des cibles 1).....	138
Fig. VI.30 : Temps mis par les deux robots pour atteindre les cibles (emplacement des cibles 2).....	139
Fig. VI.31 : Temps mis par les deux robots pour atteindre les cibles (emplacement des cibles 3).....	139
Fig. VI.32 : Scènes avec emplacement robots-cibles.....	139
Fig. VI.33 : Temps mis par les trois robots pour atteindre les cibles (emplacement des cibles 1).....	140
Fig. VI.34 : Temps mis par les trois robots pour atteindre les cibles (emplacement des cibles 2).....	140
Fig. VI.35 : Temps mis par les trois robots pour atteindre les cibles (emplacement des cibles 3).....	140
Fig. VI.36 : Scènes avec emplacement robots-cibles.....	141
Fig. VI.37 : Temps mis par les trois robots pour atteindre les cibles (emplacement des cibles 1).....	141
Fig. VI.38 : Temps mis par les trois robots pour atteindre les cibles (emplacement des cibles 2).....	141
Fig. VI.39 : Temps mis par les trois robots pour atteindre les cibles (emplacement des cibles 3).....	141
Fig. VI.40 : Progression de la durée au fil de l'expérimentation.....	144
Fig. VI.41 : Durée maximale et minimale par expérience jusqu'à ce que la victime soit trouvée.....	144
Fig. VI.42 : JetBot/AlMustakshief3D.....	145
Fig. VI.43 : Le modèle 3D du nouveau LiDAR.....	145
Fig. VI.44 : Segmentation ROI SLAM, sur RVIZ et Gazebo.....	145

Liste de tableaux

Tab. I.1 : Types et Caractéristiques des réseaux IoT.....	17
Tab. II.1 : Comparaison générale.....	55
Tab. VI.1 : Dépendance mutuelle des fonctions membres.....	126
Tab. VI.2 : Temps de simulation d'un robot pour retrouver les deux victimes.....	142
Tab. VI.3 : Durées moyennes de deux robots sans algorithme de collaboration.....	142
Tab. VI.4 : Durées moyennes de deux robots avec algorithme de collaboration	142
Tab. VI.5 : Durées moyennes de trois robots sans algorithme de collaboration.....	143
Tab. VI.6 : Durées moyennes de trois robots avec algorithme de collaboration.....	143
Tab. VI.7 : Durées moyennes par expérience.....	143
Tab. VI.8 : Temps de simulation pour retrouver les deux victimes.....	143

Liste des algorithmes

Algorithme IV.1 : L'algorithme glouton direct.....	93
Algorithme IV.2 : L'algorithme génétique pour l'allocation de tâches.....	95
Algorithme IV.3 : L'algorithme Simulated Annealing.....	96
Algorithme VI.1 : La MRTA, Algorithme de recherche orienté glouton.....	131

Problématique

Problématique

Depuis plus d'une vingtaine d'années, les avancées dans la production de machines robotisées (Fig) ont bouleversé beaucoup de domaines d'application réelle [1]. Dans l'industrie comme dans le domaine des services publics, les machines développées, dotées de faculté robotique et de capacité de traitement des données [2], attirent les attentions de manière grandissante. L'industrie de la micro-électronique et des processeurs dotés d'intelligence artificielle embarquée [3-7] (Nvidia y a pensé en embarquant de l'IA dans un GPU-RTX, pour le convertir en NPU), ont changé tous les critères sur quoi, un processus de gestion adéquate des systèmes industriels doit être équipé, tout en ayant un gain sur le compromis performance/prix. Les besoins et champs d'application des sociétés humaines sont devenus plus en plus complexe et diversifiés, ainsi l'adjacence des lieux où sont exercées ces activités, s'avère encore plus accentuée.

Ce chevauchement est d'autant plus prononcé que les tâches et les exercices humains sont liés. Le champ d'utilisation de ces machines s'est diversifié pour toucher spécifiquement des tâches d'intervention sur des sites industriels/isolés, où la présence d'opérateurs humains est risquée [8, 9].

A cet effet, faire intervenir un robot sur un site définit hostile à la présence humaine, réduit considérablement les coûts des opérations. L'intervention par un robot autonome, risque d'augmenter les coûts, afin d'améliorer ces charges, on procède par supervision guidée par opérateur humain. Cette technique, a fait intervenir un champ de recherche dit COBOTIQUE.

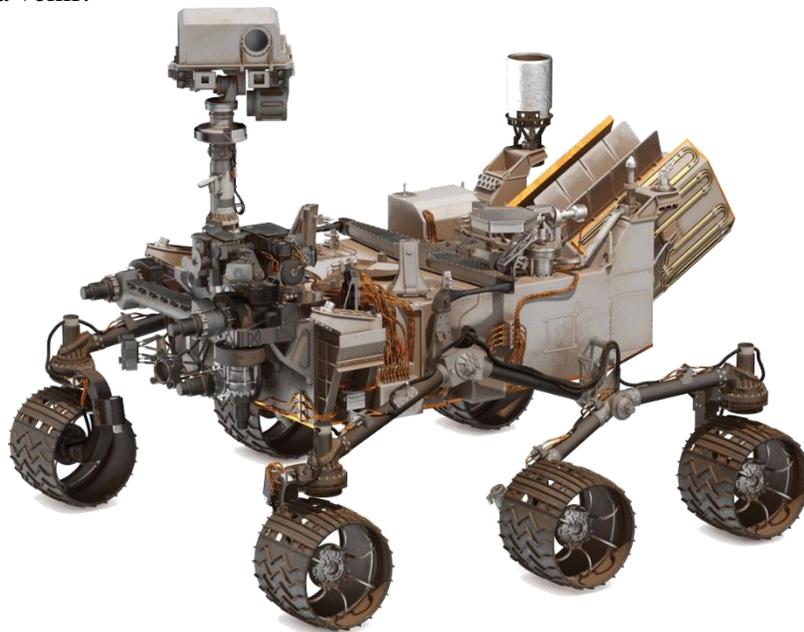
Les recherches dans ce sens, tentent à améliorer les performances des opérations menées en partie par des opérateurs et des robots en même temps. Un tel exercice, risque de provoquer des questions relatives à la sûreté des évolutions machines robots sur un lieu et des opérateurs humains.

La question centrale à laquelle il faut répondre, selon cette thèse, est relative à l'amélioration des coûts endurés d'une manœuvre en cas d'intervention sur un lieu-dit *hostile* tout en ajustant de manière adéquate les aptitudes opérationnelles [10]. Ces missions sont fonction de plusieurs impositions pour lesquelles, une perte de contrôle, peut entraver une quelconque intervention.

Certes, nous comptons contribuer selon une approche collaborative, répondre aux questions suivantes :

- Définir le contexte systémique et ses aléas
- Proposer une modélisation globale
- Délimiter les interactions et les échanges, en termes de données
- Cartographie et Identification du contexte hostile sur lequel nous intervenons
- Planifier les éventuelles tâches (selon conditionnement du site hostile)
- Collecter le maximum d'information durant l'exécution des tâches et les transmettre indirectement par un tiers moyen et reporter les incidents.
- Répondre avec moins de ressources possibles, aux exigences de la mission.
- Réduire la consommation de l'énergie en rationalisant les tâches
- Permettre l'interopérabilité des moyens et l'améliorer.
- Permettre, tant que possible la reconfigurabilité des équipements pour une mission suivante.

Les informations détaillées concernant ces points seront fournies dans les sections à venir.



*Fig : Véhicule ROVER M2020, Crédit NASA/JPL-Caltech
Véhicule conçu pour des missions spatiale sur MARS*

Introduction générale

Introduction générale

Dans ce travail de recherche, notre thèse représente une contribution qui se reporte à l'introduction de machines robotisées (de faible empreinte, grande efficacité) pour effectuer des missions d'intervention fonctionnelle sur un lieu hostile. Le caractère majeur de ces lieux représente une difficulté certaine sur la présence d'opérateur humain. Cette thèse discute deux situations extrêmes pour lesquelles, ces machines peuvent être autonomes, dotées de capacités *intelligentes* ou *supervisées* à partir d'une console derrière laquelle un opérateur humain peut envoyer des instructions et collaborer conjointement avec la machine en action sur les lieux. Ces machines interviennent en formation cluster (un petit groupe) et travaillent (autonomes/supervisés) selon un plan de collaboration désigné par une stratégie globale, dite COLLABORATIVE.

Du point de vue algorithmique, le contexte de cette thèse, s'intègre dans des cas de travaux de recherche MRTA (Multi-Robot Task Allocation) [11-15]. Chaque élément du groupe est un AGENT, et est équipé de moyen lui permettant d'effectuer une tâche dépendant de ce qu'il porte comme charge utile (MCU (Algorithmes), Camera, LiDAR, Batterie, IMU, Telecom, Motricité).

Pour se faire, le robot agent est selon l'un des mode autonome ou supervisé, commandé ou autorisé à exécuter des tâches à distance alors que l'opérateur participe, de manière collaborative, à l'envoi desdites commandes, chose qui évite un contact physique direct avec l'environnement de l'exercice durant une certaine manœuvre. Des techniques sont utilisées et sont basées sur la multiplication de ces machines robotisées sur un même lieu dans le but d'effectuer une même tâche, ces techniques offrent de grands avantages en termes de gain de temps et d'efforts mais la prise en compte de plus d'un robot sur un même lieu risque de compliquer la situation globale si ces machines fonctionnent en mode autonome ou selon une stratégie supervisée [16-18].

La sécurité du système (dans notre cas basé sur des techniques IoT) est au centre des conception et de la topologie choisie et étudiée. Pour des raisons de la conception du système, ces agents microrobots sont montés en *cluster* et sont déployés selon deux visions complémentaires :

1. Le déploiement logique : topologie n'ayant trait qu'à la structure du réseau en termes de répartition des traitement algorithmiques effectués sur la totalité des machines (qui seront prises statiques ou mobile) [19].

2. Le déploiement physique : topologie qui traite les rapprochement physique matériels en relation avec la qualité de la machine (robot ou simple machine de traitement automatique) [20].

Enfin, un traitement de fond, relatif à la collaboration a été lancé. Ces techniques sont relatives aux problèmes SMA-MRTA [21] et utilisent des algorithmes et approches pouvant aider, plus ou moins, la notion de collaboration multi-agents [22, 23]. L'approche adoptée prouve la justesse de notre proposition.

Dans cette même thèse, nous avons développé des méthodologies, des codes de calcul et contribuer à redévelopper un moyen d'acquisition des données 3D des lieux à examiner (le *vLiDAR3D*, version à breveter) - Brevet à soumettre à INAPI pour approbation.

Pour la problématique énoncée, le concept collaboration fait surface et exige une intervention particulière sur ce dont les robots sont dotés et équipés de charge utile et d'une logique embarquée sur leurs unités de traitement. Plusieurs configurations sont offertes en tant que base de déploiement d'une structure de cluster composé d'agent microrobot. La morphologie du groupe peut être centralisée autour d'un Sink (station centrale dans le réseau) ou décentralisée pour laquelle chaque agent est doté de capacités matérielles (charges utiles) et logiques (algorithmes). Les configurations sont diverses, leurs utilisations varient selon l'approche à adopter en tant que stratégie de gestion du système multi-agents (Multirobots) et qui reste à valider selon des scénarios et études de cas.

La question centrale dans ce contexte est de déterminer la localisation spatiale des microrobots et la répartition optimale des micro-tâches afin qu'ils puissent exécuter leurs missions. A remarquer que le but de leur intervention est de mener à bien une tâche globale, répartie en micro-tâches. Cette localisation spatiale ainsi que la répartition des micro-tâches sont imposées par des obstacles physiques (non finis à priori), entravant un mouvement direct sur une trajectoire (à prédire) pour atteindre un objectif donné/supposé. Un grand domaine de recherche en robotique est né, pour donner suite aux avancées technologiques et de recherches, Les recherches orientées MRTA (Allocation de tâches pour des structures multirobots)[12]. Cette technique prouve son efficacité pour des configurations décentralisées ou centralisées. En cette recherche, des algorithmes basés sur la topologie des graphes, A*, D*, Greedy ou AMAS alors qu'en parallèle d'autres algorithmes centrés sur des techniques bio-inspirés (Essaims), évolutionnistes (GA) ou récemment des idées dont l'énergie est pompée de l'IA (GPT-MAS) comme l'algorithme CrewAI

[24]. Certes, l'adoption de stratégie basée sur des agents influence nettement le traitement des tâches en offrant à chacun d'eux plus de faculté d'opérer indépendamment des autres des situations conflictuelle tout en répondant à une connectivité permanente, tant que possible, avec le sink.

La connectivité des agents, de point de vue des échanges d'informations, respecte une topologie de réseau [25]. Les agents microrobots sont plus tenus à deux configurations :

La configuration physique, les microrobots sont considéré, chacun comme un agent robot mobile sur un contexte géographique délimité, caractérisé par une complexité spécifiée physiquement.

La topologie réseau, selon le concept réseau, les agents forment un réseau d'équipements répartis (délocalisés). Cette structure réseau, peut être prise selon une approche IoT adoptant par exemple un moyen de connexion XBee ou Cloud pour les déportations de segments réseaux et ainsi pour les périphériques qui y sont connectés. La transmission des données est vitale pour la validation de la bonne exécution des tâches assignées.

La meilleure stratégie pour mettre en évidence une optimisation des missions que le cluster de micro-cobots se charge de régler, c'est une fonction multi-objectifs. Certes, il est demandé d'effectuer un ensemble de tâches dans un temps réduit, tout en augmentant la robustesse du système vis-à-vis de l'utilisation des ressources embarquées. Il est clair que ces derniers imposent mécaniquement parlant, le véhicule durant son évolution sur le lieu. Cette charge utile, consomme de l'énergie, principalement délivrée par une batterie à bord.

La participation dans la mission en tant qu'élément du cluster, n'a de sens que l'agent sélectionné est en mesure d'être membre actif pour le groupe et par rapport à la nature de la mission à accomplir [26]. A cet effet, cette thèse est organisée de la manière suivante :

Une problématique, durant laquelle, un questionnement est livré pour mettre en clair le bon fondement du thème choisi. C'est aussi, l'ensemble des questions relatives à une réalité remarquée pratiquant une difficulté dans les situations de gestion de tâches sur des lieux définis comme hostiles.

Une introduction générale succédera pour mettre en exergue, la requête centrale, définit par la contribution au développement d'un cluster de micro-cobots pour l'intervention sur les sites hostiles. En cette phase, nous passerons en revue la littérature scientifique en encadrant les spécifications de la question et son environnement par rapport aux approches mathématiques décrites dans différentes recherches et publications.

Au chapitre premier, nous introduisons une section réservée aux définitions étymologiques en rapport avec le contexte de la thèse tout en explorons par une suite d'encarts réservés à la théorie de la cobotique et ses applications ainsi que la théorie de topologie des graphique et son rapprochement au problématiques relatives à l'affectation et programmation des tâches. Parlant, topologie, il sera question de subdiviser la contribution en deux parties, l'une pour la répartition des ressources matérielles étant donné que nous manipulant un cluster de microrobots sur un site géographique et l'autre pour l'aménagement de stratégie basée sur le concept algorithmique de recherche pour l'utilisation des ressources pour améliorer l'efficacité de la collecte de données nécessaire pour l'exécution des tâches (missions de chaque microrobot dans le groupe).

Dans le chapitre suivant, une modélisation est présentée en guise de premier pas vers l'approche solution. En ce chapitre, les deux volets feront l'objet de propositions systémique. Nous présenterons les modèles de classes (Structuration logique de la question et de l'approche solution proposée ainsi que l'algorithmique nécessaire) et de déploiement (Répartition et relation technique entre les équipement mis en adéquation pour la réalisation effective d'une quelconque solution).

La sécurité du système IoT est le sujet majeur du suivant chapitre où il est mis en évidence, le positionnement de la question et la méthodologie de veille à respecter pour des mesures de sécurité. Ce critère est une exigence du bon fonctionnement du système. Certes, les agents du même cluster de microrobots travaillent dans un synchronisme établit. Des connexions radio manipulant des tonnes de données relatives au site à examiner et/ou à la mission et ses recommandations.

La modélisation est suivie par un autre chapitre réservé à la mise en action des modèles adoptés. L'étude de cas, sera un chapitre de mise en pratique des modèles proposés. Nous définissons ; la stratégie, le scénario (Plan d'action et Objectif) et nous suivront par la discussion de la simulation pratiquée. L'environnement de simulation est un contexte bien connu dans la littérature, c'est le système ROS avec ses outils GAZEBO, RViz et son ensemble de packages.

Une discussion générale est entamée suite aux résultats de la simulation sous environnement ROS. Durant la conclusion générale, nous discuterons le positionnement de la problématique par rapport aux objectifs de solution tracés tout en comparant nos résultats selon l'approche adoptée par rapport à d'autres solutions selon la vision matérielle ou logique.

Chapitre I
Etude théorique

Chapitre I : Etude théorique

I.1. Introduction

Les inventions qui suscitent un vif intérêt font rapidement le tour des médias, propulsant la science-fiction et initiant des chefs-d'œuvre vers un avenir lointain, la recherche scientifique de solutions s'inspirant de ces avancées bouscule les frontières et élimine au plus que possible les entraves. Trop tôt, les recherches, remarque que l'usage de tels machines dans des situations précises avec une présence humaine, aura des effets sensibles à la sureté des opérateurs. En effet, Isaac Asimov crée les célèbres quatre lois de la robotique, bien qu'initialement mentionnées au nombre de trois. Asimov, le forger du terme, ajoute une loi cruciale dans sa conclusion, appelée la "loi zéro". Ces règles fondamentales stipulent qu'un robot ne peut nuire à l'humanité ni permettre, par son inaction, que l'humanité soit exposée à des dangers. De plus, un robot ne peut porter préjudice à un être humain ni, en restant passif, permettre qu'un être humain soit exposé au danger, sauf en contradiction avec la loi zéro. En outre, un robot doit obéir aux ordres d'un être humain, sauf s'ils entrent en conflit avec la première loi, la loi zéro, ou la deuxième loi. Enfin, un robot doit protéger sa propre existence tant que cette protection ne contredit pas la première ou la deuxième loi, ainsi que la loi zéro [27, 28].

Les années 50 sont principalement marquées par diverses œuvres de science-fiction qui introduisent leurs propres robots, tels que Robby le robot, qui connaîtra une longue carrière cinématographique et dont le concept sera repris dans plusieurs créations [29]. Ces robots envahissent les écrans et le monde, avec des films parfois plus obscurs comme Robot Monster [30]. Sur le front des robots réels, on assiste à l'avènement des "renards électroniques" du chercheur *Albert Ducrocq* (Fig.I.1). Parmi eux, le cinquième de cette série, surnommé Job le renard, se démarque. Grâce à divers capteurs censés reproduire la version basique des cinq sens, Job le renard était capable de se diriger, de contourner des obstacles et même d'apprendre de ses erreurs. Ces informations étaient stockées dans une mémoire, certes limitée, mais présente sous forme de bande magnétique. Bien que représentant une avancée significative, ces machines demeurent analogiques, limitées dans leurs fonctionnalités et dans le développement d'une véritable intelligence artificielle autonome [31].



Fig.I.1 : le renard de Ducrocq

L'intelligence artificielle (IA) suscite un intérêt croissant parmi les chercheurs, comme en témoigne l'article [32]. Dans cet article, Turing introduit le concept qui est maintenant connu sous le nom du test de Turing. En substance, ce test évalue la capacité d'une machine à imiter un être humain à un point tel qu'un interlocuteur humain ne puisse pas déterminer, au cours d'une conversation, s'il interagit avec un autre être humain ou une machine dotée d'intelligence artificielle [33].

I.2. Développement des robots

Un test que les chercheurs vont essayer de passer avec succès durant des décennies. L'arrivée du premier circuit intégré en 1958 va cependant permettre de grandes avancées dans le domaine technique en général, y compris le développement de la robotique et de l'intelligence artificielle. En parallèle, arrive en 1961 une nouvelle autre grande avancée, l'arrivée du premier robot industriel de l'histoire c'est *Unimate* (Fig.I.2), un robot qui est intégré aux lignes d'assemblage de *General Motors* [29, 34]. Il s'agit d'un bras manipulateur, articulé capable de transférer un objet d'un endroit à un autre et inspiré des téléopérateurs utilisés dans l'industrie nucléaire dans les années 1950 pour la manipulation d'éléments radioactifs [29].

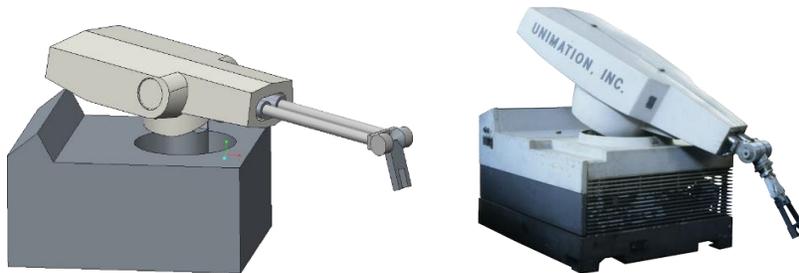


Fig.I.2 : Le robot Unimate de Unimation-General Motors

Dans les années 1970, grâce aux progrès des circuits intégrés VLSI et les applications connexes, les robots ont commencé à remplacer les humains pour certaines tâches, spécifiquement celles que les humains ne pouvaient physiquement pas accomplir en raison de l'agressivité de l'environnement. La conquête de l'espace avec, par exemple, Lunokhod 1 (Fig.I.3) et 2, les

premiers véhicules Rovers partiellement autonomes envoyés sur la Lune en 1970 [35] et d'autres modèles de robots comme dans la figure I.4.



Fig.I.3 : Linokhod 1, engins spatiale Russe, envoyés sur la lune

Autre évolution majeure des années 70, l'arrivée du premier robot bipède capable de marcher. Ancêtre des robots *humanoïdes* modernes, celui-ci était capable de voir, de communiquer et même de saisir des objets. Avec cette progression de l'explosion technologique, ces types de machines robotisées vont devenir de plus en plus rapide et plus fiable en termes de mouvement, de traitement de l'information et d'interaction avec les environnements autour. Les humanoïdes se dotent de capacités esthétiques leur permettant d'imiter les expressions faciales et d'entretenir des conversations de plus en plus fluides. De nos jours, des modèles robot comme *Erica* la présentatrice robot qui aurait dû faire la présentation d'un journal télévisé au Japon. Outre les *humanoïdes*, les robots d'une manière générale, sont aussi devenus présent au même titre que l'informatique dans divers domaines de la société humaine et sur les lieux où sont pratiquer les activités métier, surtout celles présentant une difficulté par rapport à une présence humaine.

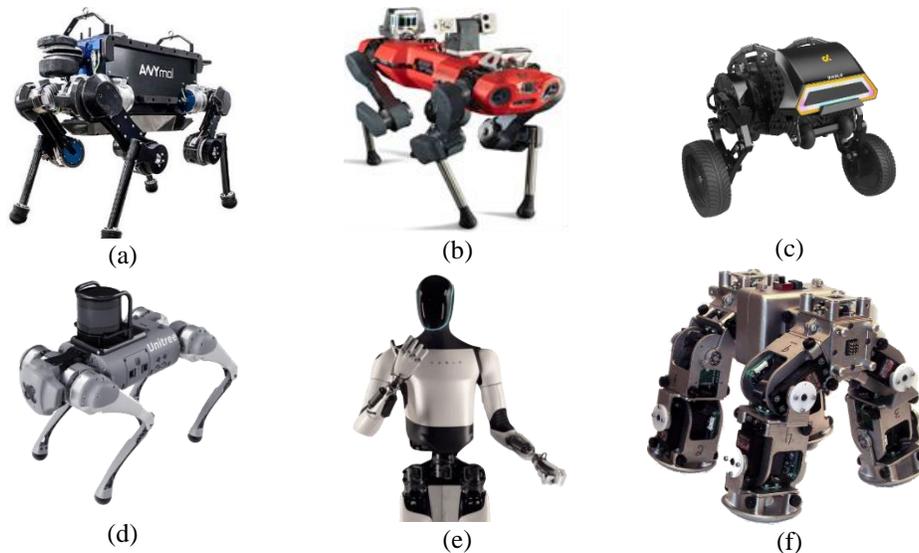


Fig.I.4 : le robot (a) ANYmal émanant du laboratoire ETH, (b) ANYMAL C d'ANYBOTICS et (c) Diablo robot de Direct Drive Tech., (d) UNITREE Cheetah, (e) TESLA Optimus & Humanoïde, (f) Sel-Reconfiguring Robot

A présent, les recherches se dirigent en particulier vers la prospection et la production de micro et nano robots, Le domaine de micro/nano robotique est le résultat de réflexions visant rationaliser la charge utile (Matériel embarqué pour l'acquisition/l'interaction avec le milieu hostile) avec les tâches pour laquelle une machine robotisée pourrait être sélectionné comme moyen, tout en réduisant la consommation de l'énergie et optimisant l'efficacité durant la supervision ou l'échange de données. Ces machines sont prises dans un groupe afin de se partager la tâche globale. C'est le contexte de notre approche, comme par exemple pour agir sur les cellules, véhiculer des traitements médicaux de manière très précise pour des applications en nano-médication, intervenir dans des lieux trop compliqués et restreints (cas des ruines historiques ou bien dans des immeubles touchés par des tremblements de terre). Les *MEMs** sur la figure I.5.a [36], les *NANObots* sur la figure I.5.b [37] ou les nanorobot à *GRPHENE* comme dans la figure I.5.c [38] sont des exemples concrets de l'évolution du domaine de la nano/microrobotique. Les nano/micromachines peuvent être dotés d'aptitudes intelligentes tout en adjoignant (logique embarquée ou via Sink, sur le Cloud) des algorithmes intelligents. Les technologies en rapport avec les avancées en Télécommunication, offrent une variété de moyens, à embarquer, pour des fonctions d'échange ou de transmission des données.

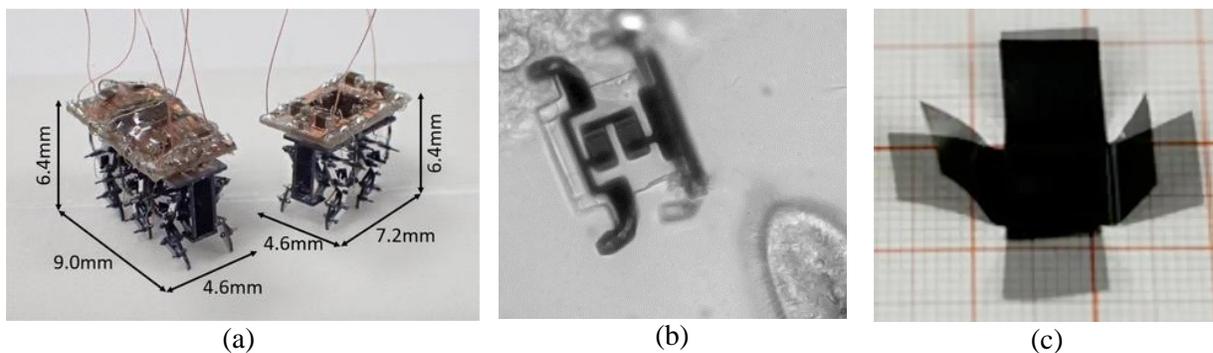


Fig.I.5 : Robots de dimension NANO, (a) Robot *MEMS*¹ à Réseaux NN, multipède contrôlé par IC avec mécanisme indépendant*, (b) Nanorobot activé par Laser* (c) Nanorobot par *GRPHENE**

Du point de vue de l'algorithmique embarquée, les machines robotisées renferment, des unités de traitement dotées de CPU+GPU de capacité équivalente et fonction de la mission. Les systèmes d'exploitation à bord sont personnalisés pour offrir à ces machines légèreté, portabilité et interopérabilité selon les missions et la charge utile. Pour la raison d'offrir le caractère autonomie,

¹ Micro Electro-Mechanical Systems

les microrobots portent des algorithmes basés sur des stratégies de collaboration dont les fondements sont tirés des techniques de l'IA et de l'optimisation du matériel en termes de consommation énergétique (Fig.I.6). L'intelligence artificielle a évolué à un tel point que le test de *Turing* a finalement été passé avec succès pour la première fois en 2014, un robot utilisant l'intelligence artificielle ayant réussi à se faire passer pour un enfant de 13 ans. De cette manière, les observateurs disent que la réalité a rattrapé la science-fiction ou bien la science-fiction a aidé la réalité à se développer. À l'heure actuelle, des approches de type *Gen AI*, peuvent être embarqué sur des microrobot, lui offrant ainsi des capacités hors normes de traitement et d'interaction avec le milieu. Ces techniques dérivant de la nouvelle version de l'IA, utilisent des modèles neuronaux multi-étages dits LLM, basés sur le langage, avec des interfaces de codage/décodage (Transformers). Ces modèles ayant acquis des connaissances suite à des traitements d'apprentissage faisant digérer des milliards de faits codifiés dans une base de données. Un raffinement des connaissances, peut être effectuer de manières spécifiques à un domaine définit. Ces boites à outils, interagissent avec leur entourage, utilisateur humain ou système machine, de manière simple et efficace, basée sur un langage prédéfini pour la communication. Les modèles *LLM*, génèrent des réponses sur la base de requête soumissionnée. Des plateformes *GenAI* tel que *CrewAI*, utilise une stratégie répartie sous forme d'agent communiquant, solidaires dans un groupe (*Crew*) pour exécuter à bien une tâche donnée. Ces derniers se partage, intelligemment la tâche globale et interagissent si une limitation est atteinte par l'un d'eux.

Ces modèles *LLM* acquièrent des compétences énormes, du traitement de la parole, du texte, la voix, l'image et même la vidéo. Ces compétences formeront un plus, une fois montées sur des microcontrôleurs des agents microrobots. *Sophia* (Fig.I.7), premier robot humanoïde complètement connecté au réseau et ayant reçu des faveurs juridiques pour la première fois, fait référence à une machine, dont l'inférence à quoi est tirée ses interactions et ses pseudo-émotions, est le résultat de modèle IA adaptatifs.

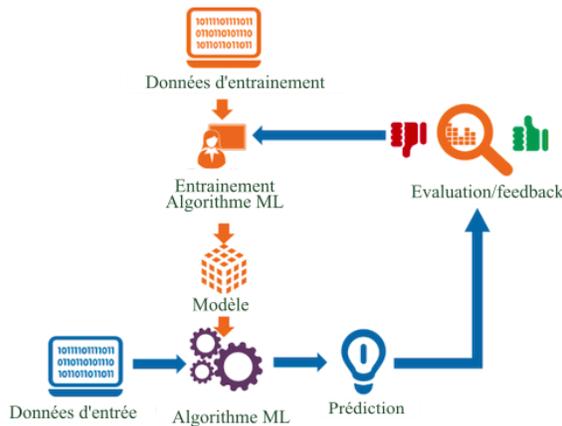


Fig.I.6 : Structure de préparation d'un LLM



Fig.I.7 : Génoïde Sophia de Hanson Robotics

Sophia, le robot à l'apparence humaine le plus avancé de Hanson Robotics, incarne nos rêves pour l'avenir de l'IA. Combinaison unique de science, d'ingénierie et d'art, Sophia est à la fois un personnage de science-fiction inventé par l'homme et représentant l'avenir de l'IA et de la robotique, et une plateforme pour la recherche avancée en matière de robotique et d'IA' (Site Web HANSON Robotics). Sophia est le premier robot au monde ayant reçu une reconnaissance civile et une nationalité.

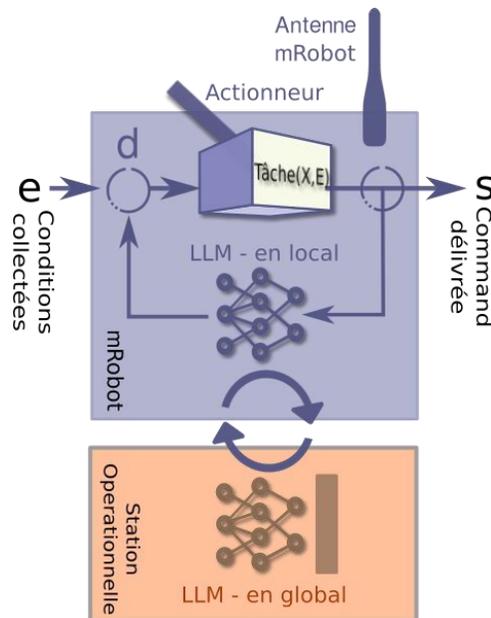


Fig.I.8: MRTA with LLM Based control

A nos jours, on distingue mal la différence entre robotique et IA, une fois que l'un est la complémentarité obligatoire pour l'autre en 2024. Un processus logiciel orienté agent, peut

naviguer dans le réseau (c'est le cas d'agent mobile MAS), ou un microrobot physique doté de facultés GenAI exécutant des tâches (Fig.I.8). La seule différence dans ce cas, c'est la manière avec laquelle on interagit avec l'environnement. A nos jours, une question d'ordre réglementaire surgit impliquant ainsi un questionnement juridique sur les limites de responsabilité en cas d'implication de machine robotique dans des situations où la 'loi ZERO' a été ignoré.

I.3. Les critères relatifs à la morphologie et à la mobilité

L'une des difficultés à résoudre, pour un microrobot, est le pouvoir de mobilité et l'évolution sur des trajets définis dans des environnements hostiles, tout en inhibant les entraves et les causes des dégâts, La dernière observation est fonction de beaucoup de paramètres du microrobot intrinsèques et/ou extrinsèques. De la capacité de se prendre en charge durant une mission aux complexités de l'environnement dans lequel ils évoluent. Le facteur de forme, définit une empreinte de la machine microrobotique, de telles sortes que les facultés d'intervention peuvent en être directement impliquées. On prend en considération la partie forme durant les opérations de design et celle de l'autonomie énergétique pour la motorisation. En ce qui concerne la partie morphologie, le système doit être adapté le plus possible à la nature géographique difficile du milieu à étudier. À titre d'exemple, le robot-serpent d'exploration en milieu hostile (Fig.I.9a), ce robot se meut de façon ondulée en imitant parfaitement les mouvements d'un serpent, conçu par l'institut robotique de l'Université Carnegie Mellon de Pittsburgh (USA). Il grimpe, s'enroule le long d'un poteau. Il peut même nager, passer à travers des canalisations et s'enfiler autour de divers obstacles. Doté de capteurs d'images et de lasers, il peut scanner en trois dimensions le paysage, le visage et les objets. Il est contrôlé à distance par une radiocommande. Mais le robot-serpent, utilisé en cluster, pourra aussi servir à la manipulation d'engins explosifs ; chaque agent agissant comme un doigt ou un tentacule pour déplacer l'objet [39].



Fig.I.9 : Sortes de microrobots, (a) robot-serpent [39], (b) robot bug [40]

Beaucoup d'autres exemples et avec des morphologies bio-inspirées (Fig.I.9b) sont utiles pour la manœuvre sur des milieux hostiles (Fig.I.10).



Fig.I.10 : Un robot éliminant les obstacles dans les égouts [41]

Pour la partie motorisation, le pouvoir de se déplacer dans les zones hostiles avec aisance est une condition pour les microrobots mobiles (nécessité des manœuvres dans des sites hostiles). La mobilité, chez les robots, en général, peut avoir des façons multiples.

Selon leur mobilité, les robots sont classés comme selon les catégories suivantes :



Fig. I.11 : Robot à 4 roues

[42]



Fig. I.12 : Robot à chenilles

[43]

- Par des roues (Fig.I.11) : La structure surfacique d'une roue peut diffère selon l'environnement, ils fournissent une souplesse de déplacement, l'énergie appliquée moins que d'autres moyens de déplacement.
- Par des chenilles (Fig.I.12) : Ce type consomme trop d'énergie, ce moyen offre mécaniquement, une stabilité de la machine.
- Mono/Multipède (Fig.I.13) : Meilleur façon et la plus adaptée aux trajets sur sol et surtout si l'environnement est caractérisé par une topographie hostile, ce type de motorisation est difficile à réaliser ainsi que à programmer, si l'on considère les drivers moteur qui équipent et font bouger les pieds de manière fluide et synchrone.



Fig. I.13 : Robot Multipède [44]



Fig. I.14 : Robot à mobilité hybride [45]



Fig. I.15: Les petits robots Kilobots

- Hybrides (Fig.I.14 et Fig.I.15) : Il est possible qu'un robot peut se déplacer en utilisant une combinaison de manière pour le déplacement.

Le choix d'un type de motorisation, dépend du moyen pour la mobilité, du degré de liberté DDL qu'offre chaque configuration et de la complexité de l'environnement de l'étude, ce qui nécessite une révision mathématique du conditionnement du déplacement chez les robots/microrobots. Certes, la Mécanique Rationnelle et la Modélisation Cinématique des Robots, font le traitement mathématique de la mobilité chez les robots en général. Le cadre de notre thèse, se limite à cette partie comme hypothèse afin de définir la topologie et la structure, sur quoi la mobilité est basée. Si l'on considère qu'un robot, est un système mécanique articulé et est une somme d'articulations (*chaîne cinématique*), selon une première vision de la topologie des graphes. Les membres joints sur une articulation donnée, font l'objet d'un rattachement lie une partie du robot à une autre. Si on définit, un point P_i , représentant le bout en extrémité de ces membres, ses coordonnées par rapport à un repère cartésien (orthogonal, orthonormé $\mathcal{R}(O, x, y, z)$) sont (x_1, x_2, x_3) où \bar{X} . On peut écrire les relations :

$$\bar{X} = F(q) \text{ Correspondant au modèle cinématique directe,}$$

$$q = G(\bar{X}) \text{ Correspond au modèle cinématique inverse}$$

Il faut dire que chaque articulation, forme mathématiquement une relation bilatérale et une contrainte d'égalité qui s'ajoutera au système d'équations d'une configuration robot. Nous nous intéressons, seulement aux modalités d'un configuration mathématique finie et à base de quoi nos simulations sont entreprises. Dans notre thèse, les caractères cinématiques de nos microrobots, du point de vue des manœuvres, sont de type à $\delta_M = \delta_s + \delta_m$ où δ_M est le degré de Manœuvrabilité DoM , δ_s le degré Dos de maniabilité (Steerability) et δ_m le degré de Mobilité DoM (à savoir que $\delta_M \leq DOF$, DOF ou DDL étant le degré de liberté (Fig.I.16)).



Fig. I.16: Structure générale d'un robot à 2 DDL (Rotation+Translation)

Nous nous focalisons sur une configuration donnée à 4 roues (roues avant à 3 DDL), portant une charge utile d'équipements nécessaires pour effectuer, à bien la mission, tout en estimant une collaboration, induite par échange et transmission de données entre les agents d'un même cluster de microrobots.

I.4. La robotique et les télécommunications

En tant que domaine scientifique, les communications ne se limitent pas à l'étude des antennes et des signaux. Il s'agit plutôt d'une science technique qui englobe tout ce qui peut être transmis à distance par des moyens techniques, et la manière de le faire de manière efficace et compétente. La science des communications, qu'elle soit câblée ou sans fil, se concentre principalement sur le traitement de l'information. Cela comprend la conception de l'information à transmettre. Elle comprend également l'extraction du message et la correction des erreurs à la réception.

Les robots généralement nécessitent une supervision à distance par un opérateur, ils utilisent aussi des capteurs pour l'interaction avec leur environnement, Les microrobots montés en cluster restent des machines qui nécessitent de l'interaction et doivent échanger les informations qu'ils possèdent avec d'autres et afin que la collaboration se concrétise. Cet échange (Fig.I.17) d'information relative à l'état du groupe et de son entourage, requiert le moyen télécommunication (Tab. I.1).

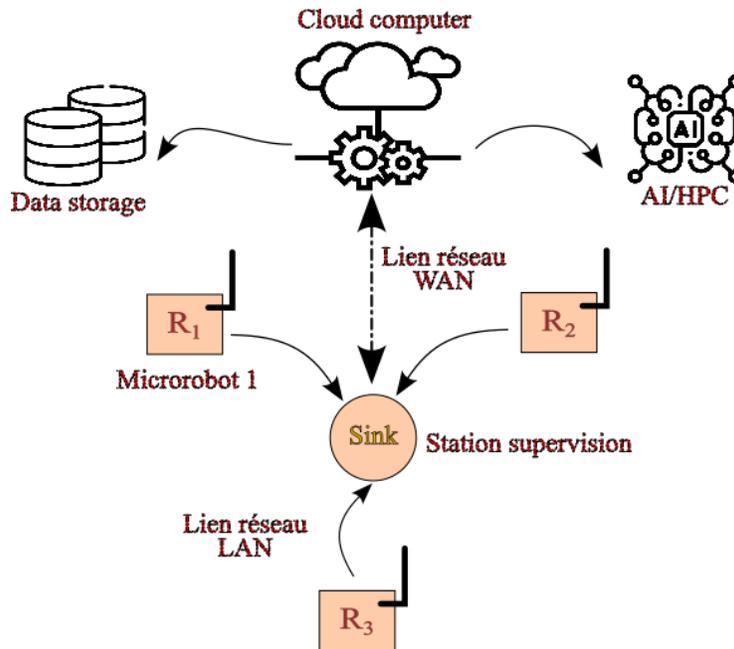


Fig. I.17 : Mise en réseau IoT du cluster de microrobots

Tab. I.1 : Types et Caractéristiques des réseaux IoT

Caractéristique\Types	Cellulaire	LAN/PAN	LPWAN	Mesh Protocols IWMP
Vitesse	~100Kb/s-100Mb/s	~100Kb/s-100Mb/s	~10Kb/s	~250Kb/s
Largeur bande	Longue	Courte	Longue	Courte
Longévité batterie	Moyenne	Moyenne	Longue	Longue
Utilisation	Traditionnel M2M	Domotique	Projet IoT large	Projet IoT large
	Communication traditionnelle Agriculture Intelligente	Périphérique IoT portée Wifi BLT Domotique	Localisation GPS Ville intelligente Visio Conférence	Gestion de l'éclairage Visio Conférence Conditionnement

I.5. La cobotique

On dit cobot ou robot collaboratif, c'est exactement un robot conçu co-exister avec un être-humain pour interagir physiquement de manière garantissant la sécurité des deux parties. La collaboration est un travail de groupe synchrone de plusieurs agents dans la réalisation d'une tâche commune [46]. La cobotique est une discipline émergente qui s'intéresse en particulier à l'interaction, soit directe ou téléopérée, entre système robot et opérateur humain pour accomplir une mission/tâche commune [47]. Les systèmes hommes/robots collaboratifs, appelés systèmes Cobotiques [48], co-existent avec différents types d'interaction, différents moyens de perception, des conditionnements d'une efficacité, modalités d'action, un besoin de prise de décision et leur intensité sont au cœur des problématiques de ce domaine. La cobotique et la robotique sont presque

proches. La cobotique n'englobe cependant pas tous les défis de la robotique, car l'intérêt de l'étude n'est pas le robot en lui-même, qui peut avoir des performances particulières en fonction des caractéristiques électroniques, énergétiques, automatiques ou mécaniques ce qui a priori n'a pas beaucoup d'importance ici. D'autre part, symétriquement, la robotique ne renferme pas tellement la cobotique, car quelques aspects, comme l'ergonomie du poste de travail, la représentation que l'humain se fait du robot ou l'acceptabilité par l'opérateur, dans le cadre d'études ergonomiques et de l'ingénierie cognitive.

La cobotique renferme également des questionnements et des études, qui concernent typiquement les interactions hommes robots, comme par exemple le degré et l'efficacité du couplage, ou la dynamique spatio-temporelle de l'activité étudiée par le système cobotique pris comme une entité unique. D'un point de vue spécifique, la cobotique est donc à l'intersection de trois grands domaines, l'ergonomie, la cognitive et bien sûr la robotique. Ses problématiques avec l'IA, des facteurs humains et de l'interface homme machine, ils sont tous également proches.

Le domaine de la cobotique industrielle englobe les systèmes de téléopération, de colocalisation collaborative, de co-manipulation et d'exo-manipulation (robots revêtus par l'homme, désignés exosquelettes actifs ou encore orthèses robotiques) [49].

L'INRS, ou Institut National de Recherche et de Sécurité, est une institution française dédiée à la prévention des accidents du travail et des maladies professionnelles. Fondé en 1947, l'INRS joue un rôle crucial dans la promotion de la sécurité et de la santé au travail en France. Il a une vision singulière de la cobotique (Fig.I.18). Il classe les dispositifs techniques selon leur application, industriel, de service ou médicale [50]. Pour les concevoir, les systèmes cobotiques peuvent être classés suivant leur type de fonctionnement ou suivant leur domaine d'application.

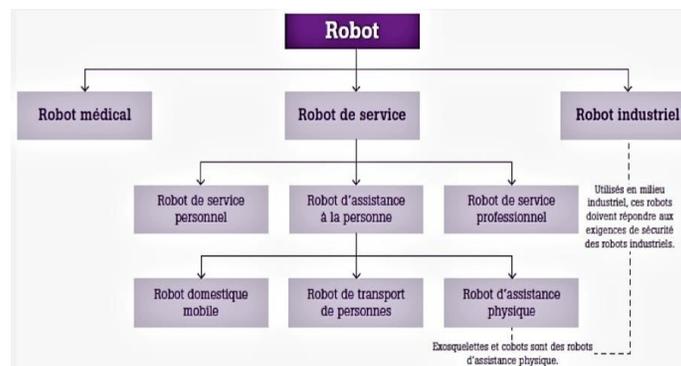


Fig. I.18 : Classification des robots selon l'Institut National de Recherche et de Sécurité [49, 50]

Les trois entités dans la figure I.9 (le système robot, l'opérateur et la tâche) ont des propriétés matérielles (hardware, physiques) et immatérielles (software). Le procédé de fabrication, les outils et la pièce par exemple constituent la partie matérielle de la tâche, alors que, sa composante logicielle contient les objectifs du système, formellement définis ou pas. Le système robot est également dual [51], une partie électromécanique (le hardware) et une autre, le programme informatique et les données (le software).

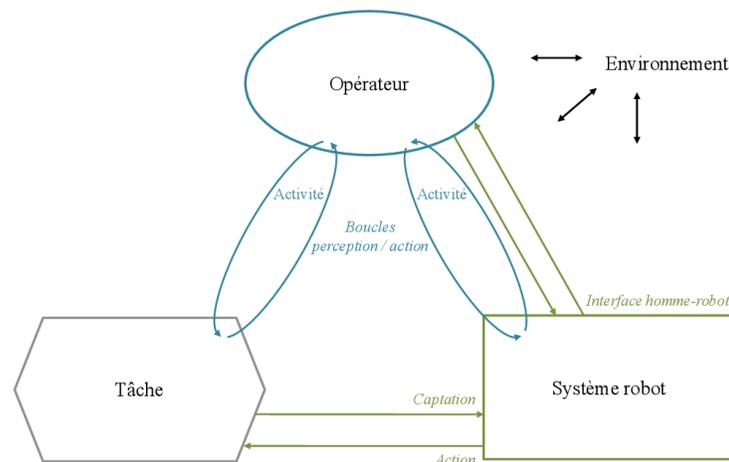


Fig. I.19 : Schéma général d'un système cobotique [49]

L'homme, soit au travail ou non, est un corps physique renfermant une cognition morale (intelligence), donc même l'opérateur n'échappe pas à la dualité matériel/logiciel de *David Bitonneau* [51]. Les propriétés matérielles et immatérielles ne sont pas indépendantes. Pour la tâche, l'objectif définit le procédé et réciproquement pour tout système électronique pure ou électronique hybride comme les robots qui sont des électromécaniques, le software ordonne les actions au matériel et ce dernier rend possible et conditionne le software. Finalement, pour l'opérateur, le corps est interdépendant à la cognition dans ce qui est appelé par Varela (1996) la cognition incarnée.

I.5.1. Naissance de la cobotique

En ce qui concerne le terme cobot, il a été introduit pour désigner des robots industriels collaboratifs travaillant avec l'humain dans le même espace de travail [52]. Aujourd'hui, cette définition est souvent admise par les roboticiens, surtout dans le milieu industriel. Les drones sont rarement appelés cobots. Cependant, du point de vue des interactions des systèmes avec les êtres humains et de la perception (cognitive), cette restriction du champ coopératif (on peut dire

uniquement industriel) et du mode de coopération en collaboration (dans le même espace de travail) paraît arbitraire. On préfère de ne se limiter pas seulement à la robotique industrielle, On englobe tous les robots collaboratifs par une définition élargie. La définition d'un robot qui soulève quelques difficultés, pour lever l'ambiguïté et la distinction dans certains systèmes mécaniques de robots, propose de définir le terme robot comme un système mécatronique constitué d'un ensemble de capteurs, d'un objet tangible, d'un microcontrôleur programmable et d'un système moteur équipé d'actionneurs de manipulation ou de mouvement. Compte tenu de la définition d'un cobot comme robot collaboratif, il faut noter qu'une description physique abstraite n'est pas suffisante pour déterminer la nature d'un robot. Cela nécessite de déterminer s'il collabore ou non. Un robot conçu pour la collaboration n'est donc qu'un cobot potentiel. La difficulté est d'identifier, par définition, une voiture comme un cobot [53].

La cobotique dérive du concept de cobot. Le terme a été introduit en 1996 par Colgate, Peshking et Wannasuphprasit [54], au sein de l'université North western (États-Unis) et du département d'ingénierie mécanique. Le cobot, comme mot, est une combinaison issue entre les mots robot et collaboration, désigne un dispositif passif (C'est-à-dire, l'homme est celui qui fait une énergie fournie pour les déplacements) dont l'actionnement permet de contraindre et guider les mouvements d'une personne, en définissant des surfaces virtuelles programmables.

Le premier prototype de robot industriel a été développé en 1998 en collaboration avec General Motors [55] pour aider les opérateurs à décharger les portes des voitures (Fig. I.20).



Fig. I.20 : Un déchargeur de porte au sol développé par l'Université Northwestern est testé dans une usine GM [55]

Depuis 2008, une nouvelle étape a été franchie pour vulgariser les interactions homme-robot dans l'industrie avec la commercialisation du premier robot collaboratif, UR5 (Fig. I.21a), par la société danoise Universal Robots. Après les années qui ont suivi, les principaux fabricants de

robots industriels ont développé leur modèle de « robot coopératif », la société allemande Kuka qui a commercialisé LBR iiwa (Fig. I.20b) en 2011.

Toujours en 2011, la société française Robotiques 3 Dimensions a commercialisé son premier robot d'assistance collaboratif, approchant le concept initial du cobot UR5 aux chercheurs de la Northwestern University. Ce robot manipulateur, qualifié de "cobot" par Robotiques 3 Dimensions [56], est capable de programmer pour compenser le poids ou l'inertie des charges manipulées, pour diriger les gestes des opérateurs à l'aide de surfaces virtuelles et pour multiplier les efforts d'un homme [49].

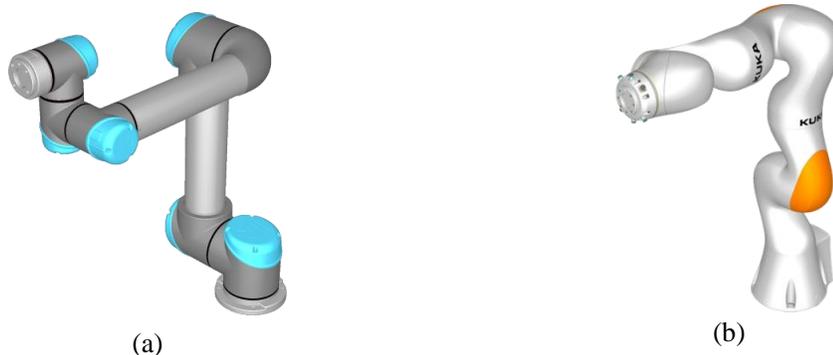


Fig. I.21 : L'UR5 et LBR iiwa pour (a) et (b) respectivement

En ce qui concerne la sécurité des dispositifs et face à l'émergence de ces nouveaux types d'appareils, un grand nombre d'organismes (publics, privés, centres de recherche, instituts ou entreprises) se sont réunis pour normaliser leur intégration dans les postes de travail individuels et leur construction. Cela, en 2011, a conduit à la publication de la norme internationale « AFNOR : NF EN ISO 10218-1/2 » pour l'exigence de la sécurité des robots industriels pour les intégrateurs des robots industriels et les concepteurs, pour décrire les nouveaux principes de fonctionnement à savoir la collaboration homme-robot dans un milieu partagé. La cobotique ne se limite pas à l'étude et à l'utilisation des cobots. Elle porte au sens large sur la collaboration homme-robot, que ce soit avec des robots manipulateurs mobiles [57] ou des exosquelettes (squelette externe) [47]. Cette collaboration peut s'appuyer sur des interactions directes ou à distance. Pour l'industrie, la cobotique est une opportunité pour perfectionner la performance des entreprises, ainsi que la santé et la sécurité des personnes sur les postes de travail. Elle offre une nouvelle façon de transformer des tâches inefficaces, ennuyeuses ou dangereuses, lorsque l'automatisation ancienne n'a pas pu remplacer l'expérience des opérateurs ou lorsqu'elle n'était assez rentable.

Les robots peuvent aider les humains à réduire l'effort physique nécessaire à l'exécution d'une tâche ou à les placer dans un environnement en toute sécurité si c'est qualifié comme environnement hostile. De plus, en ce qui concerne une cellule automatisée, le système robot peut être clarifié par l'utilisation des savoir-faire, des capacités de perception, d'adaptation et de prise de décision de l'homme. La cobotique se trouve, pour des raisons commerciales, entre le travail manuel et l'automatisation pour un volume de production assez grand et reste le meilleur choix pour un volume petit et moyen comme illustré sur la figure ci-dessous (Fig. I.22). La figure met en valeur l'apport économique potentiel de l'utilisation des cobots avec des humains par rapport au travail manuel, à l'utilisation des robots et à l'utilisation de machines spéciales (machines dédiées pour fonctionnement aux des tâches spécifiques et qui sont limitées en termes de mouvement, articulation). Le travail manuel peut être préférable pour des très faibles volumes de production (<100 unités produites), alors que les systèmes classiques sont généralement plus rentables pour des productions en très grande série, avec pas trop de variations et des durées de cycle faibles. Pour des volumes et cadences de production intermédiaires, on peut considérer que la cobotique puisse apporter une augmentation de la rentabilité et un gain en flexibilité.

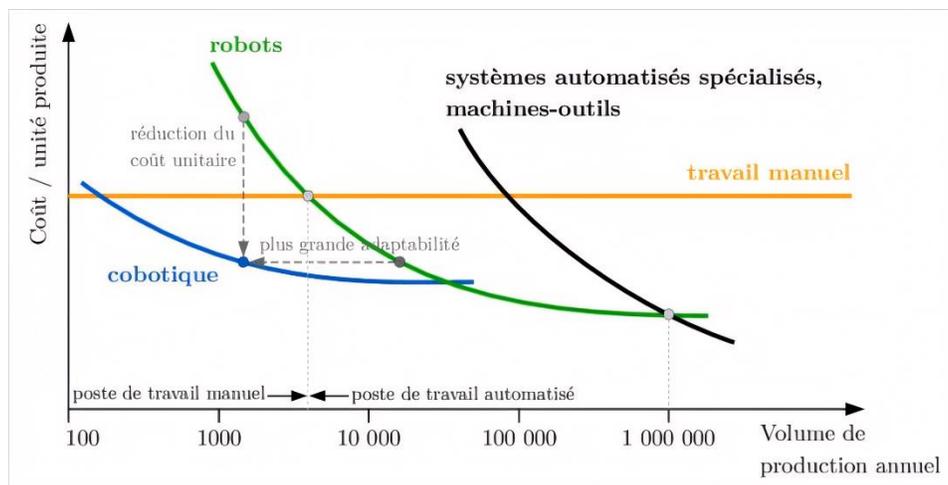


Fig. I.22 : Coût par unité de production en fonction du volume de production annuel pour la cobotique, comparé au travail manuel, aux robots et à l'automatisation, adapté de Makarov [58] d'après Hägele et al [59]

I.5.2. Classification en fonction du rôle de l'opérateur

Les applications de la cobotique sont nombreuses comme la diversité des systèmes cobotique. De pas mal d'auteurs et chercheurs ont essayé de classifier ces systèmes pour structurer le domaine

de la cobotique [60, 61], même des communautés spécialisées dans les problématiques sociales des interactions hommes machines [62], ou dans les problématiques physiques et sécuritaires de ces mêmes interactions [63, 64].

On peut regarder aux robots et les distinguer suivant [53] :

- Leur structure architecturale.
- Leur capacité d'autonomie.
- Leur pouvoir de mobilité et manipulation.
- Leur taille.
- Leur éloignement de l'opérateur.

Dans ce cas, si on remarque, la classification omet totalement la composante humaine qui est indispensable dans le système cobotique. En revanche, Scholtz a pu classer les systèmes cobotiques en fonction du rôle de l'humain sur leur proposition [65] :

- L'opérateur : Il (télé)opère le cobot, qui a généralement une autonomie décisionnelle très faible, comme le drone à cause de sa difficulté de le mettre en intégralité avec l'humain.
- Co-travailleur : Il travaille parallèlement avec le cobot dans un environnement commun et il y a parfois quelques interactions avec lui, par exemple en lui prenant un autre chemin de mouvement si quelqu'un dedans.
- Superviseur : Il fournit des instructions au cobot en vérifiant éventuellement la qualité du travail. On peut noter que les instructions doivent être fournies alors que le cobot est en fonctionnement.
- Spectateur (Observateur) : Il existe dans le même endroit que le robot et entre parfois dans la zone de travail du robot, mais pas avoir de réelle interaction avec lui. Le robot a, parfois et comme exemple, un capteur de présence qui lui indique qu'un être humain est dans sa zone et il passe automatiquement dans un mode plus lent ou s'arrête pour une période de temps. La collaboration est réduite au strict minimum, mais on peut malgré tout parler de cobot car le mode opératoire dépend bien, de manière indirecte, de l'humain.
- Opérateur de maintenance : La cobotique n'est pas concernée par la maintenance.
- Concepteur/programmeur : La cobotique n'est pas concernée par la programmation du robot.

La classification, on peut dire qu'elle est une peu restrictive pour l'utilisation dans l'industrie. On ne peut pas toujours dire qu'un humain est un opérateur en termes de rôle. Selon le domaine d'application, il peut s'agir d'un n'importe quelle personne soit d'une passion, visiteur, d'un militaire, d'un enfant, à titre d'exemples. Cette classification distingue implicitement le type d'interaction qui existe entre un cobot et un humain. Mais, elle reste relativement superficielle parce qu'elle ne permet pas de prendre en compte le degré d'autonomie du cobot, pour distinguer par exemple l'exosquelette et le robot télécommandé. Afin de classer les systèmes cobotiques, il faudrait prendre en compte les principales caractéristiques des interactions, tout en intégrant des informations robotiques ou humaines si elles sont essentielles. Il est bien sûr possible de multiplier les critères descriptifs et de les associer pour établir une classification plus pertinente, mais cela perd alors en simplicité [53].

I.5.3. Autonomie des cobots

Les recherches dans l'intelligence artificielle ont fortement amélioré la capacité des robots à résoudre des problèmes très complexes, ce qui est mis en stade la cause d'augmenter leur autonomie et a découvert plus de modalités d'interaction [66, 67].

Des progrès ont été réalisés dans des domaines réputés complexes, comme la reconnaissance des formes et de la parole, la navigation en terrain non-identifié/hostile ce qui n'est, en général ; qu'un traitement de l'information acquise, les erreurs et les incertitudes sont encore essentiels pour se proposer des mises en œuvre effectives, simples et rapides pour des applications non triviales. Les approches de type Deep Learning (l'apprentissage en profondeur), associées à des systèmes multi-agents et méthodes symboliques sont prometteuses, mais la complexité des problèmes cycliques nécessite la prise en compte de connaissances implicites et d'informations tirées du contexte, ce qu'on ne fait pas efficacement [67, 68]. Selon *Lemaignan*, ces difficultés structurent la recherche autour de trois défis : la communication homme-robot, l'action conjointe et l'explicitation de l'intention [69]. Ces principes sont généralement basés sur une architecture de contrôle qui permet une planification fine des tâches, un haut niveau d'abstraction, ce qui influence la collaboration.

I.5.4. Cobotique et ses applications

Il n'est pas facile de citer les applications tant elles sont diverses et nombreuses. Sheridan le spécialiste dans le domaine des facteurs humains, a proposé ; en 2016 ; une revue de synthèse à partir de 4 applications relativement représentatives des travaux que peuvent être en collaboration hommes robots [70], qui sont :

- **Contrôle supervisé** : L'humain ici, son intervention est de faire commander et surveiller les actions du robot, par exemple pour faire une digitalisation 3D téléguidée à un emplacement donné, effectuer une tâche très sensible exige une précision en commandant le robot, etc. Une certaine autonomie est nécessaire sur chaque tâche.
- **Commande à distance** : il s'agit typiquement du cas des UAVs ou drones, que ce soit aérien, terrestre ou sous-marin, et généralement les tâches sont non triviales et peuvent être dangereuses. En général, il y a peu d'autonomie à cause de la nécessité de minimiser les erreurs, toutes les commandes viennent de l'humain.
- **Véhicule autonome** : l'interaction de l'humain, ici, est juste comme un passager dont le rôle se résume à définir une destination, par exemple le mode pilote automatique dans un avion, ou bien comme les voitures électriques Tesla auto-conduite.
- **Interactivité sociale** : c'est le cas du robot conçu pour rendre des services ou bien a la capacité d'existence dans un milieu avec une communauté humaine telle quelle est, par exemple le robot d'assistance aux personnes âgées.

Cette taxonomie n'est pas représentative de la variété de toutes les applications des cobots, mais permet déjà de distinguer de grands ensembles, et on peut considérer que cette thèse, en premier lieu, focalise sur la deuxième et la dernière application.

I.6. Collaboration multirobots

La collaboration est essentielle pour une performance efficace des groupes de robots dans les contextes d'intervention en cas de tâches difficiles, pénible ou spécialement en cas de catastrophes. Le développement récent des technologies robotiques a introduit des robots dans divers domaines de l'industrie, de l'agriculture, de la sécurité, etc. Cependant, les applications complexes nécessitent plusieurs robots pour exécuter une tâche de manière efficace, par exemple, la manipulation d'un objet lourd ou l'assemblage d'un produit complexe. Ainsi, un grand effort de recherche a été fait au cours des trois dernières décennies sur le contrôle coordonné de plusieurs robots.

Les systèmes multirobots peuvent être définis comme un groupe d'agents robotiques coordonnés afin d'effectuer une tâche particulière, qui peut ne pas être réalisable par un seul robot ou qui peut être optimisée si elle est effectuée par un groupe [71]. Cela signifie qu'une analyse de la tâche est nécessaire afin de déterminer s'il faut ou non utiliser une collaboration multi-robot. Cette analyse devrait se concentrer sur la recherche d'une éventuelle augmentation de l'efficacité obtenue lors de l'utilisation d'un MRS au lieu d'un système robotique unique pour une tâche donnée [72]. Cette section contextuelle présente une synthèse des classifications du système multi-robot selon des caractéristiques ou des typologies, en tenant compte de la taille de l'équipe, du niveau de coopération, de la morphologie et du type de tâche réalisée. La taxonomie et les définitions mentionnées proviennent d'études de plusieurs auteurs [73-75].

I.6.1. Classement par taille de l'équipe

L'une des classifications les plus simples pouvant être effectuées dans un système multirobot est basée sur le nombre d'agents composant le système. Dans ce cas, quatre groupes peuvent être identifiés :

- Unité unique bien que techniquement il ne s'agisse pas de système multirobot, on peut dire que le nombre minimum d'agents est un.
- Système à deux unités est le nombre minimum de robots requis pour former une équipe. Deux équipes d'unités se concentrent généralement sur des tâches qu'un seul robot ne peut accomplir seul ou pour obtenir des capacités complémentaires.
- Les systèmes multi-unités font référence à des équipes composées de plusieurs membres mais où le nombre d'unités est relativement petit par rapport à la zone/tâche requise. Il s'agit peut-être du type d'équipe le plus courant, et donc avec les types de robots, de stratégies et d'applications les plus variés.
- Les essaims (Swarms) fait référence à une équipe de robots dans laquelle le nombre de membres est très important, voire illimité. Habituellement, les robots en si grand nombre ont des capacités limitées et atteignent leurs objectifs grâce à des comportements émergents. De plus, ce type d'équipes doit être très résistant aux changements du nombre de membres.

De plus, en ce qui concerne la taille de l'équipe, un classement différent peut être effectué. Ne prenant pas en compte le nombre de membres de l'équipe mais la capacité de l'équipe à se réorganiser lorsque ce nombre change. Dans ce cas, il existe trois catégories différentes.

- **Disposition statique** : La configuration est fixe et la tâche ne peut être réalisée en cas de perte d'un des agents. Il s'agit du cas le plus simple et il s'agit généralement de la configuration des systèmes à deux unités.
- **Réaménagement coordonné** : Dans ce cas, il est possible de modifier le nombre de robots dans l'équipe. Mais tout changement de ce type nécessitera une étape intermédiaire, comme une renégociation ou une reconfiguration de la mission. Par conséquent, chaque robot se voit attribuer une nouvelle tâche ou un nouvel ensemble de tâches.
- **Disposition dynamique** C'est la configuration la plus complexe. Le nombre de robots peut changer et la mission doit se poursuivre sans aucune réaffectation des tâches. Cela signifie que système multi-robot doit détecter de manière autonome qu'un de ses membres est parti et adapter dynamiquement le comportement des membres restants pour atteindre l'objectif principal. C'est le cas le plus courant des essaims de robots. Il est également utilisé par certains systèmes multi-unités, mais il peut entraîner une sous-tâche d'agencement supplémentaire très complexe.

I.6.2. Classification par morphologie et capacités des robots

La morphologie d'un robot fait référence à différents aspects : La description mécanique, le type de locomotion et la cinématique du robot. Chacun de ces aspects, combiné à la puissance de calcul et de traitement du robot, définit les capacités du robot. Cela signifie qu'une classification des robots ou systèmes multirobots selon leur morphologie impliquera intrinsèquement leurs capacités. Sachant cela, il est possible de définir trois catégories différentes.

- *Identique* : Les membres du système multirobot sont homogènes tant dans leur mécanique que dans leurs capacités. Cela ne signifie pas que tous les robots effectuent la même tâche, mais plutôt que toutes les tâches de la mission peuvent être assignées à n'importe lequel des membres de l'équipe et qu'ils peuvent être remplacés par n'importe quel autre membre de l'équipe.
- *Homogène* : Les membres du système multirobot ont une morphologie ou des capacités similaires mais pas exactement les mêmes. Cela signifie que les membres de l'équipe peuvent effectuer la même tâche dans une certaine mesure. Alors que certaines capacités sont exclusives à un sous-

groupe limité de l'équipe. Par exemple, une équipe de robots à roues qui ont certains capteurs en commun mais pas tous pareils. Pour résumer, cela signifie que si l'un des membres de l'équipe est supprimé, la mission ne peut être que partiellement accomplie.

- *Hétérogènes* : Les membres hétérogènes du système multirobot diffèrent à la fois par leur morphologie, leurs capacités ou leurs techniques de locomotion. Des exemples typiques sont les systèmes multirobots constitués de robots aériens et terrestres. Contrairement aux catégories précédentes, la mission ne peut être accomplie si l'un des membres est retiré.

I.6.3. Classification par niveau de coordination entre les systèmes multirobots

La plupart des travaux fondateurs dans ce sens ont proposé des algorithmes de contrôle centralisés, basés sur des informations globales par rapport à un système de coordonnées commun. En particulier, les systèmes de commande centralisés sont efficaces dans la commande de mouvement coordonnée des manipulateurs à base fixe, car le nombre de robots en coordination est généralement limité à deux ou trois. Cependant, les progrès récents des manipulateurs mobiles, qui permettent le mouvement libre dans un environnement réel, ont considérablement augmenté le nombre de robots pouvant être impliqués dans une tâche coordonnée. Ainsi, les approches centralisées deviennent irréalistes, en raison de la charge de calcul et du fait que diverses erreurs géométriques qui apparaissent inévitablement parmi les robots ne peuvent pas être traitées avec précision sur la base d'un système de coordonnées commun. Pour surmonter les problèmes susmentionnés, un contrôle décentralisé de plusieurs robots est apparu, dans lequel chaque robot est contrôlé par son propre contrôleur basé sur son propre système de coordonnées local [76]. Parlant sur la centralité et topologie du réseau en terme détachages de données, il est considéré deux dimensions des topologies de réseau : cosmopolite/provincial et centralisé/décentralisé. Des exemples sont présentés sur la figure ci-dessous (Fig.I.23).

Dans la dimension cosmopolite/provinciale, un robot communiquant avec un autre robot crée un lien entre eux. Dans cette dimension, la distance entre les robots détermine la force de ce lien. La distance plus courte indique un lien plus fort. Pour distinguer les liens forts et faibles, il est utilisé ; dans l'illustration ci-dessous pour (Fig.I.23.a) et (Fig.I.23.b) ; des lignes fines pour représenter les liens faibles tandis que les lignes épaisses sont utilisées pour représenter des liens forts. Les réseaux provinciaux contiennent principalement des liens forts et redondants et très peu de liens faibles tandis que les réseaux cosmopolites sont constitués de nombreux liens faibles et

très peu de liens forts [77]. Pour la topologie centralisée/décentralisée, les réseaux centralisés ont quelques nœuds clés avec des emplacements ou des fonctions importants dans le réseau. Au contraire, dans les réseaux décentralisés, chaque nœud a une importance similaire. Dans cette dimension, la structure entièrement centralisée est toujours appelée réseau centralisé, où tous les nœuds se connectent à un nœud central (comme le montre la figure I.23.c)).

Contrairement au réseau entièrement centralisé, le réseau avec la topologie entièrement décentralisée est appelé un réseau distribué (comme le montre la figure I.23.e). Dans un réseau distribué, les nœuds ont la même importance en termes de position, de distance aux autres nœuds, de nombre de liens, etc. [78].

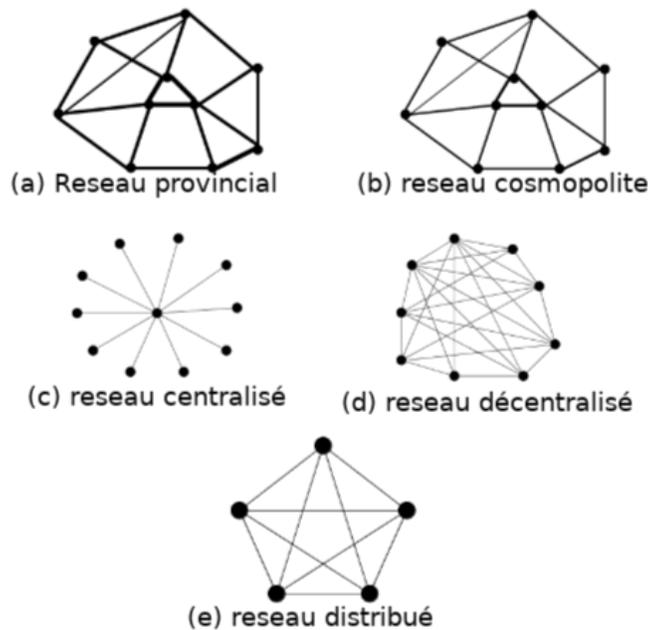


Fig. I.23 : Des exemples de réseaux provinciaux et cosmopolites sont présentés dans la première ligne ((a) et (b)) de cette figure. (c), (d) et (e)) sont respectivement un réseau entièrement centralisé, décentralisés et entièrement décentralisés (distribués) [78]

La classification [79] définit différents niveaux de coordination et d'interaction entre les robots d'un système multi robot. Ce type de classification a été largement étudié et les catégories présentées ici ont été utilisées dans [80, 81]. Un résumé de la classification par trois niveaux de coordinations. Le premier niveau de séparation est basé sur le fait que les robots soient conscients ou non qu'ils coopèrent au sein du système multi robot. Même si les robots d'une équipe ne se

connaissent pas, il est possible d'obtenir des tâches coopératives, ou du moins coordonnées. En revanche, lorsque les robots se connaissent, trois sous-groupes différents sont définis selon leur niveau de coordination. Dans ce cas, la coordination fait référence à la capacité d'un robot à prendre en compte les actions exécutées par les autres robots. Lorsqu'il y a coordination, elle peut être de deux types, forte ou faible selon que la coordination est implicite ou non, c'est-à-dire qu'elle suit un ensemble de règles de coordination prédéfinies (fortement coordonnée) ou qu'elle obtient une coordination émergente ou réactive (faiblement coordonnée). Enfin, l'organisation du système multi robot est également à prendre en compte. Cette différenciation repose sur la capacité de chaque membre à prendre des décisions autonomes sur les actions à réaliser. Là encore, on retrouve trois catégories différentes [82] : la coordination distribuée, où chaque robot décide de sa tâche de manière autonome ; des systèmes fortement centralisés, où les décisions sur les actions de tous les robots de l'équipe sont prises par un seul leader ; et faiblement centralisé, où les systèmes multirobots ont des architectures hiérarchiques centralisées localement ou où il n'y a qu'un seul agent qui prend la décision, mais le rôle de leader peut changer d'un robot à l'autre à différents moments ou situations.

I.6.4. Architecture logicielle pour les systèmes multirobots

DeLoach et al. [83] ont observé que si de nombreuses architectures récentes orientées agent ont été développées, il y a eu peu de tentatives pour appliquer des approches de haut niveau à la conception de systèmes robotiques coopératifs. Alors que l'utilisation d'approches multi-agents pour la robotique coopérative peut fournir certains des éléments manquants mis en évidence dans de nombreuses applications robotiques coopératives, telles que la généralité, l'organisation adaptative et la tolérance aux pannes. Par conséquent, *DeLoach* et al. [83] ont appliqué la méthodologie Multi-agent Systems Engineering (*MaSE*) pour concevoir des comportements coopératifs de haut niveau entre des robots autonomes et hétérogènes pour des applications de recherche et de sauvetage. *MaSE* [84] est un processus en sept étapes à usage général associé à une séquence détaillée de modèles graphiques interdépendants, qui guide un développeur de système d'une spécification de système initiale à la mise en œuvre de systèmes multi-agents hétérogènes. Il fournit une approche descendante pour la construction de systèmes robotiques coopératifs au lieu de l'approche ascendante basée sur le comportement utilisé dans les implémentations robotiques traditionnelles [85]. En outre, il a été conçu pour être indépendant de

toute architecture de système multi-agent, langage de programmation ou Framework de communication particulière, ce qui le rend approprié pour implémenter des applications robotiques coopératives. Néanmoins, contrairement aux architectures robotiques standard, *DeLoach* et al. [83] se sont concentrés sur la conception uniquement de comportements coopératifs de haut niveau car ils supposaient que les comportements de bas niveau communs aux robots mobiles existaient déjà dans les bibliothèques.

Donc, pour la configuration logicielle requise pour le multi-robot coopératif selon *DeLoach* et al. L'objectif global du développement de l'architecture logicielle est de faciliter la programmation d'applications distribuées pour les robots de service coopératifs et autonomes. À titre d'illustration, considérons une équipe de plusieurs robots autonomes déployés dans un environnement intérieur (par exemple, bureau, maison) qui aident les humains dans les activités quotidiennes. Considérez également que ces robots doivent se coordonner entre eux pour toute nouvelle mission à exécuter. Par exemple, lorsqu'un utilisateur envoie une commande à l'équipe des robots pour livrer un courrier d'un bureau à un autre, les robots doivent se coordonner afin que seul le robot avec le coût le plus bas exécute la mission. Bien que le scénario semble être simple, il y a beaucoup de défis à relever du point de vue de l'ingénierie logicielle.

Ces défis peuvent être énumérés dans les exigences fonctionnelles suivantes [86] :

- Télécommunication : il s'agit d'une exigence essentielle pour les applications robotiques distribuées, car les robots doivent communiquer pour pouvoir se coordonner et échanger des informations entre eux. Le problème principal de communication est la couverture sans fil limitée des robots, ce qui empêche certains robots d'être joignables par d'autres. En tant que tel, chaque robot s'appuiera sur les informations partielles qu'il obtiendra des robots voisins pour prendre une décision sur l'exécution de la mission. Par exemple, avec une couverture limitée, deux robots non voisins peuvent décider d'exécuter la même mission en raison d'un manque de télécommunication.
- Résolution de problèmes : pour toute application de robots coopératifs, un consensus doit être atteint entre les robots pour que toute mission soit exécutée. A ce titre, chaque mission représente un nouveau problème qui doit être résolu de manière distribuée dans tous les robots. Dans l'exemple ci-dessus, le problème était de trouver et de sélectionner le robot avec le coût le plus bas pour exécuter la tâche. Cela nécessite un agent spécifique dans

chaque robot chargé de trouver une solution efficace, de manière distribuée, pour toute nouvelle mission partagée entre les robots.

- Base de connaissances : dans les applications distribuées, chaque robot doit avoir sa propre base de connaissances collectant en permanence des informations sur l'environnement actuel, les autres robots et les tâches accomplies et non accomplies. La base de connaissances aide le robot à prendre des décisions optimales ou bonnes pour toute nouvelle mission à venir. Cela introduit une autre couche d'intelligence, permettant aux robots de mieux se coordonner pour exécuter efficacement les tâches. Par exemple, si un robot est connu pour avoir échoué dans l'accomplissement de certaines tâches précédemment assignées, il peut être exclu de la coordination sur de futures tâches similaires.
- D'autre part, en plus des exigences courantes en matière d'ingénierie logicielle, à savoir la modularité, l'extensibilité et la réutilisation, les quatre exigences non fonctionnelles clés suivantes pour le développement de l'architecture logicielle qui sont examinées.
- Décentralisation : tout algorithme ou protocole à implémenter doit être entièrement décentralisé (distribué) dans le sens où toute application ne doit pas s'appuyer sur un agent central (ou une unité) pour prendre des décisions, mais les décisions doivent être prises par des robots avec les connaissances partielles qu'ils auraient sur le système. Il s'agit d'une exigence fondamentale, car il n'est pas réaliste de supposer que tous les robots sont entièrement connectés en permanence les uns aux autres ou à un système central, car ils sont généralement dotés d'émetteurs-récepteurs sans fil à portée limitée.
- Tolérance aux pannes : toute application doit continuer son fonctionnement correctement même si certains robots deviennent défectueux avant/pendant/après l'exécution d'une mission. C'est en effet une conséquence naturelle de l'exigence de décentralisation mentionnée ci-dessus, qui évite le problème du point de défaillance unique. En effet, si un robot tombe en panne ou a sa batterie épuisée, d'autres robots devraient le détecter et devraient réussir à entreprendre la tâche précédemment assignée au robot mort.
- Mesure de la performance : il est important de pouvoir évaluer dans quelle mesure l'ensemble de robots a réussi à réaliser la mission globale. Plusieurs critères pourraient être

utilisés à cet effet, certains d'entre eux pouvant dépendre de l'application. Le critère peut varier d'un comportement satisfaisant à un comportement optimal.

- Hétérogénéité : dans les systèmes multirobots, il est courant de déployer des robots de différents types et marques. L'une des principales exigences de l'architecture est de prendre en charge différents types de robots, afin que le même code soit réutilisé par différents types de robots sans changements majeurs. C'est une question importante pour garantir la modularité et la réutilisation du code.

I.7. Conclusion

Le besoin d'intervenir dans ces milieux qualifiés hostiles (les figures I.24, I.25 et I.26) à la présence humaine, tels que les forêts, les ruines historiques, les lieux touchés par les tremblements de terre, les lieux potentiellement infectés ou radioactifs, les installations industrielles, etc.

Voici quelques illustrations montrent les milieux hostiles en Algérie :

- Les essais nucléaires Français dans les années 1960 continuent de tuer d'enregistrer des dégâts humains encore aujourd'hui en Algérie à Reggane spécialement (selon sputnik-news 01/2019), La radioactivité a touché ; selon les experts ; le sud de l'Europe et toute l'Afrique du Nord.
- Les sites frappés par des tremblements de terre (Boumerdes en 2003, Chlef en 1980...).
- Les ruines historiques (Casbah, Ksour du Nord, Ksar de Kenadsa et de Taghit à Béchar, Ksar de Tamentit à Adrar, ...



Fig I.24 : Séisme de Boumerdes [87]



Fig. I.25 : Les ruines de Casbah [88]

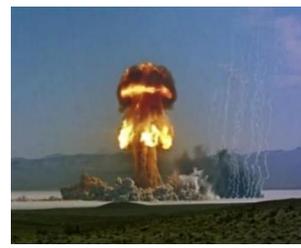


Fig. I.26 : Lieu radioactif à Reggane [89]

Ce besoin a conduit au développement de moyens et de méthodes d'action sans intervention humaine physique directe. Techniquement parlant, nous faisons allusion à la télé-opération ou à la robotique d'intervention en milieu hostile. Cette dernière utilise les robots collaboratifs, appelés communément les cobots. Dans les villes et installations industrielles, la densité des occupations

spatiales des machines robotisés et les techniciens qui exercent un suivi de routine, impose un risque majeur sur la sûreté des vies humaines autour. L'exhaustivité des informations échangées risque la perte ou l'incomplétude suite à ladite densité. Dans ce sens, les recherches actuelles se focalisent sur le regroupement de microrobots en collaboration avec des humains, d'où la notion de micro-cobots, afin de réduire la charge-utile d'un côté et la carte-de-route (tâches) de l'autre. La conception des cobots en mode micro permet aussi de minimiser la consommation totale effective de l'énergie, de réduire le prix de revient durant la conception et encore plus en cours d'une exploitation réelle.

Le but de ce travail est l'élargir notre recherche sur l'assemblage d'un cluster de micro-cobots pour la collecte supervisée et/ou intelligente des données concernant la digitalisation 3D sur les lieux hostiles cités en haut répandus sur tout le territoire national. De même, notre travail pourra s'étendre aux sites qui présentent de grands défis de supervision. La tâche principale des micro-cobots sera la collaboration pour une meilleure exécution des rôles et la transmission certaine, des données. Cette transmission utilisera tous les moyens et algorithmes de connexion disponibles dans le réseau cluster de micro-cobots. Le recours à des solutions Open-Source comme ROS (Robotics Operating System – avec ses différents modules) facilitera l'adoption d'une telle proposition. Nous notons que l'importance de la thématique de ce sujet réside dans sa nouveauté en tant que recherche universitaire et sa qualification pour l'aboutissement de la recherche dans les laboratoires de l'Université Algérienne vers des applications professionnelles de la société Algérienne.

Chapitre II

Connectivité des systèmes multirobot et réseaux de capteurs sans-fils (WSN)

Chapitre II : Connectivité des systèmes multirobot et réseaux de capteurs sans-fils (WSN)

II.1. Introduction

Les réseaux de capteurs sans fil prenant en charge la mobilité des nœuds joueront à l'avenir un rôle important. Dans les environnements résidentiels, de rééducation et cliniques, des nœuds de capteurs peuvent être attachés au corps des patients pour une surveillance à long terme et ininterrompue des signaux biomédicaux vitaux. De même, dans les environnements industriels, les travailleurs ainsi que les robots mobiles peuvent transporter des nœuds de capteurs pour augmenter leur perception et interagir de manière transparente avec leurs environnements. Néanmoins, de telles applications nécessitent des communications fiables ainsi qu'un débit élevé. Compte tenu des principaux objectifs de conception des plates-formes de détection (faible puissance, coût abordable, déploiement à grande échelle, longévité, fonctionnement dans la bande ISM), le maintien de liaisons fiables est un défi de taille. Ce défi peut être partiellement atténué si la nature de la fluctuation de la qualité de la liaison peut être connue ou estimée à temps. En effet, les protocoles de niveau supérieur tels que les protocoles de transfert et de routage reposent sur la connaissance de la fluctuation de la qualité des liaisons afin de transférer de manière transparente la communication vers des routes alternatives lorsque la qualité des routes existantes se détériore.

Les plateformes de détection sans fil émergentes promettent de jouer un rôle important dans notre vie quotidienne. Intégrant des capacités de détection, de traitement et de communication sans fil, ces plates-formes permettent une surveillance à distance, abordable, flexible et à long terme. Par exemple, elles peuvent être implantées ou portées par des patients qui peuvent se déplacer librement et effectuer des activités quotidiennes alors que des données biomédicales vitales sont étant recueilli d'eux [90, 91]. De plus, les plates-formes peuvent établir un réseau de capteurs sans fil permettant le flux bidirectionnel de données et de commandes, permettant ainsi aux médecins et assistants de santé de surveiller et d'assister leurs patients à tout moment [92-94].

Le succès de ces plateformes dépend principalement de leur capacité à établir des liaisons de communication fiables lorsqu'elles sont mobiles [95, 96]. En effet, la qualité de la liaison radio affecte presque tous les protocoles et services de niveau supérieur [97, 98]. En raison de leurs fonctionnalités à faible coût et de faible puissance, les émetteurs-récepteurs radio sont souvent sujets aux bruits de fond, aux interférences, aux évanouissements multi-trajets, aux ombres et aux

dynamiques environnementales supplémentaires telles que le mouvement des personnes et des objets environnants. En outre, des imperfections et de légères variations dans la conception et la production du matériel (telles que le placement des antennes dans les nœuds de capteurs) introduisent des irrégularités imprévisibles dans les directions de propagation radio indifférentes. Des facteurs supplémentaires tels que l'effet du corps humain et les vibrations indésirables exacerbent encore les fluctuations de la qualité des liens. Au cours de la dernière décennie, un grand nombre d'études expérimentales ont été menées afin d'étudier et de caractériser la qualité de la liaison des radios de faible puissance et la manière dont elles affectent les performances. Cependant, la plupart de ces études se concentrent sur des scénarios statiques où les nœuds sont, dans l'ensemble, stationnaires une fois déployés. En revanche, peu a été fait pour caractériser l'impact de la mobilité sur la fluctuation de la qualité de la liaison et quels facteurs doivent être pris en compte lors de la conception de protocoles MAC et de routage sensibles à la mobilité. Dans ce chapitre, nous présentons les résultats d'expériences approfondies relatives à la fluctuation de la qualité des liens dans les environnements mobiles, à la fois à l'intérieur et à l'extérieur, en se concentrant sur les mouvements à basse vitesse signifiant le mouvement des personnes dans les environnements résidentiels et de travail ainsi que des robots dans les environnements industriels.

II.2. Problèmes de la connectivité

Certains problèmes qui concernent toutes les approches traitant de la prédiction de la connectivité selon Anis Koubâa et Abdelmajid Khelil [86]. Ces problèmes peuvent être organisés en quatre aspects différents :

- Le besoin d'un système de positionnement, cet aspect comprend les approches qui nécessitent une sorte de système de positionnement, tel que le GPS ou GPS-free. Le besoin d'un GPS présente certains inconvénients, comme l'augmentation des coûts dans les implémentations à grande échelle et une consommation d'énergie plus élevée. De plus, il ne convient que pour les scénarios extérieurs. Les systèmes GPS-free ont surmonté certaines de ces limitations, car ils ont été conçus pour être utilisés avec des scénarios en intérieur. Cependant, ils nécessitent des antennes supplémentaires telles que la différence de temps d'arrivée (TDOA pour *Time Difference of Arrival*), ou des dispositifs comme Cricket Compass [99] ou même des nœuds d'ancrage. Les nœuds d'ancrage représentent des nœuds fixes dont la position est connue à l'avance. Ils diffusent leur position en

inondant le réseau de ces informations, afin que les autres nœuds mobiles puissent calculer leur propre position en utilisant la triangulation.

- Le besoin d'un journal des données historiques est un aspect qui concerne toutes les approches qui nécessitent un journal des données historiques. Un tel journal de données peut être composé de nombreuses variables différentes, telles que la position des coordonnées ou la qualité de la liaison (par exemple RSSI pour Received Signal Strength Indicator, LQI pour *Link Quality* Indicator ou SNR). Le journal des données d'historique est normalement enregistré pendant un mode hors ligne pour être traité et utilisé plus tard. Les approches qui nécessitent un historique des données sont des représentations de modèles contenues dans l'historique pour remplir le modèle. Ainsi, après le temps de traitement hors ligne, on s'attend à ce qu'il ait le même comportement pendant le temps de collecte. Le principal inconvénient de cette approche concerne les modifications possibles du motif. Si cela se produit, un nouvel historique des données est requis.
- La différence entre l'estimateur de la qualité de la liaison et le prédicteur de la qualité de la liaison, y compris également sa relation dans le contexte des réseaux de capteurs sans fil. Un estimateur de qualité de liaison (LQE pour Link Quality Estimator) est utilisé pour estimer la fiabilité du canal sans fil. Le calcul LQE peut être effectué par la carte sans fil, ce qui donne des informations telles que RSSI, SNR et LQI. Il peut également être représenté par le débit de réception des paquets (PRR, Packet Reception Rate), qui est effectué lorsque le paquet est reçu. D'un autre côté, un prédicteur de qualité de liaison (LQP) estime la valeur future à partir de la qualité de liaison. En d'autres termes, il essaie de deviner l'état futur de la liaison sans fil.
- Le type de modèle choisi et utilisé pour effectuer l'estimation de la connectivité, Il existe trois modèles principaux largement utilisés pour effectuer des prédictions de connectivité tels que les modèles de Markov, les séries temporelles et la vitesse relative. Le modèle de Markov et les séries chronologiques ont besoin du journal d'historique des données pour calculer l'estimation. En revanche, les modèles de vitesse relative nécessitent un système de positionnement précis. Le principal inconvénient du modèle de Markov est que, selon le processus, le nombre d'états peut être très important. Par exemple, si l'état est considéré comme un nœud et que le scénario comporte un grand nombre de nœuds, la taille du modèle peut devenir si grande qu'elle peut poser des problèmes aux périphériques aux ressources

limitées. Cependant, cela dépend de la façon dont le problème est modélisé. Les séries chronologiques peuvent avoir le même inconvénient, car l'historique complet des données doit être chargé dans la mémoire pour être traité. L'utilisation de la vitesse relative nécessite un système de positionnement pour donner les informations de coordonnées afin de calculer la vitesse vectorielle à partir de chaque nœud. Les limites des modèles de vitesse relative sont liées au besoin de systèmes de positionnement, comme indiqué précédemment.

II.3. Quantification et caractérisation de la fluctuation de la qualité des liens

Les études empiriques existantes sur les caractéristiques des liaisons sans fil de faible puissance suggèrent qu'elles sont dynamiques et affectées par des facteurs temporels et spatiaux, tels que la température et les environnements environnants, entre autres. Les différences dans les environnements expérimentaux, les dispositifs et les réglages des paramètres aboutissent souvent à des observations et des conclusions différentes [98, 100-103]. Certaines des mesures utilisées par ces études pour quantifier et caractériser la fluctuation de la qualité des liens sont les suivantes :

- RSSI : c'est une mesure de la puissance du signal reçu en dBm. Il est estimé sur une période de 8 symboles ($128\mu s$) dans la puce radio CC2420 (Le CC2420 est un véritable émetteur-récepteur RF à puce unique 2,4 GHz conforme IEEE 802.15.4 conçu pour les applications sans fil à faible puissance et basse tension) [104] qui est la puce radio largement utilisée dans la communauté des réseaux de capteurs sans fil.
- Rapport signal/bruit (SNR) : il s'agit d'une mesure améliorée utilisée pour mesurer la qualité de la liaison en tenant compte du bruit de fond (de canal de transmission). Il est défini comme le rapport entre l'énergie du signal reçu et le bruit de fond (la variance de bruit du canal de transmission σ_b^2). Lorsqu'il n'y a pas de signal entrant, la valeur lue dans le registre RSSI représente le bruit de fond. Par conséquent, le SNR peut être simplement calculé comme suit (Equ.I.1) :

$$SNR [dB] = RSSI - \sigma_b^2 [dB] \quad (Equ.I.1)$$

- LQI : C'est une métrique de qualité de liaison définie dans la spécification IEEE-802.15.4 [105-107] et est une caractérisation de la force et de la qualité du paquet reçu. Les vendeurs de radio l'implémentent légèrement différemment. Par exemple, dans CC2420, le LQI est

une valeur de corrélation moyenne qui est déterminée à partir des 8 premiers symboles de chaque paquet entrant et va de 50 à 110 [108].

- Nombre de paquets d'accusé de réception (ACK) : il s'agit d'une métrique de couche liaison indiquant que le paquet transmis a été reçu avec succès par le récepteur. Cette métrique peut être extraite du côté de l'expéditeur lorsque le paquet d'accusé de réception correspondant est reçu avec succès dans un délai spécifique après chaque transmission de données.
- Taux de réception de paquets (PRR) : il s'agit d'une métrique côté récepteur et peut être calculée comme le rapport entre le nombre de paquets reçus avec succès et le nombre de paquets qui peuvent idéalement être transmis dans une fenêtre de temps spécifique pour un intervalle inter-paquets défini (IPI). Cette métrique est équivalente au taux de réception de l'accusé de réception (ARR), une métrique de couche de liaison côté émetteur si l'on ne tient pas compte de l'asymétrie de liaison. ARR est parfois appelé taux de livraison de paquets (PDR) ou taux de réussite des paquets (PSR).

Les études existantes classifient largement les liaisons sans fil à faible puissance en régions connectées, transitoires et déconnectées [103, 108]. Dans une région connectée, le PDR est généralement supérieur à 90% et la communication est considérée comme hautement fiable et symétrique. Se produisent occasionnellement. Dans une région de transition, le lien est en rafale (avec un PDR variant de 10% à 90%) et asymétrique. Dans une région déconnectée, le PDR est inférieur à 10% et la perte de paquets élevée. La plupart des liaisons pratiques dans les réseaux de capteurs sans fil prenant en charge les nœuds mobiles sont caractérisées par la région de transition.

Miluzzo et al. [109] ont étudié l'impact du corps humain sur les réseaux mobiles de capteurs sans fil de faible puissance. Ils ont mené des expériences dans trois environnements différents : grand espace ouvert, environnement urbain et bureau. Le nœud mobile en tant qu'émetteur était porté par un être humain se déplaçant à une vitesse de marche et le récepteur était déployé de manière statique. Leurs résultats ont montré que le facteur corporel a un impact significatif sur la qualité des liaisons des radios à faible puissance.

Ahmed et al. [110] ont mené une série d'expériences pour étudier les caractéristiques de liaison des réseaux de capteurs aériens sans fil. Dans leurs expériences, les nœuds TelosB ont été attachés sur deux drones qui volent à la même hauteur et un nœud a été placé au sol comme station de base. Les modèles de communication étaient bidirectionnels et peuvent être classés comme sol-aérien

(G-A), aérien-sol (A-G) et aérien-aérien (A-A). Après avoir analysé les variations RSSI en ce qui concerne la distance et le taux de perte de paquets, ils ont conclu que les liaisons A-A fonctionnent le mieux.

Xiong et coll. [97] ont présenté des études expérimentales sur les caractéristiques de liaison de la radio conforme à la norme IEEE802.15.4 dans les communications véhiculaires. Dans leurs expériences, un nœud était attaché à une voiture en mouvement comme émetteur et l'autre était déployé sur le bord de la route en tant que récepteur. Ils ont analysé les caractéristiques de la liaison en termes de RSSI, de taux d'erreur sur les paquets et de distribution des pertes de paquets en ce qui concerne la hauteur de l'antenne, la vitesse du véhicule (30 à 130 km / h) et la distance (0 à 600 m) en ligne de visibilité directe LOS et scénario sans visibilité directe NLOS. Les résultats de l'expérience révèlent que la qualité de la liaison est asymétrique et affectée par la hauteur de l'antenne et l'effet Doppler. De plus, les auteurs remarquent que la taille conventionnelle du paquet ACK est trop courte pour garantir un accusé de réception réussi. En conséquence, le mécanisme ACK peut consommer environ 38% des ressources de canal dans des scénarios sans visibilité directe pour obtenir une communication fiable.

II.4. Caractéristiques des liens mobiles

En général, les liens dans les environnements mobiles sont plus dynamiques que leurs homologues statiques [111]. De cela, on explore de plus près les caractéristiques des liens mobiles, en adoptant différentes perspectives.

II.4.1. Caractéristiques spatiales

Expérience prise par [111] pour étudier les caractéristiques spatiales de la fluctuation de la qualité des liaisons dans des scénarios mobiles, ils ont mené des expériences dans des environnements intérieurs et extérieurs. Pour chaque expérience, un nœud stationnaire a été déployé comme récepteur et un nœud mobile s'est éloigné du récepteur le long d'une ligne droite pour couvrir une distance de 20 mètres de long. L'émetteur mobile envoie des paquets à un IPI de 50 ms. Pour chaque scénario, les expériences ont été répétées 10 fois. Comme observations, la figure au-dessous (Fig. II.1) met en évidence les caractéristiques spatiales en termes de RSSI, LQI et PDR. Toutes les mesures de qualité des liens ont été collectées du côté de l'expéditeur. Les trois représentations du haut sont pour l'environnement intérieur et les autres du bas sont pour l'environnement extérieur. Les points RSSI et LQI représentés sont des valeurs moyennes de 80 paquets successifs où le PDR a été généré en accumulant des paquets ACK. Les résultats a et d de

la figure montrent les variations RSSI le long de l'axe des distances. Comme prévu, RSSI a diminué tandis que la distance relative entre l'expéditeur et le récepteur augmentait, suivant le modèle de perte de chemin (Path Loss) log-normal [112] (voir aussi [113]) :

$$RSSI(d) = RSSI(d_0) - 10\log_{10}(dd_0) + N(0, \sigma_b^2) \quad (Equ.1.2)$$

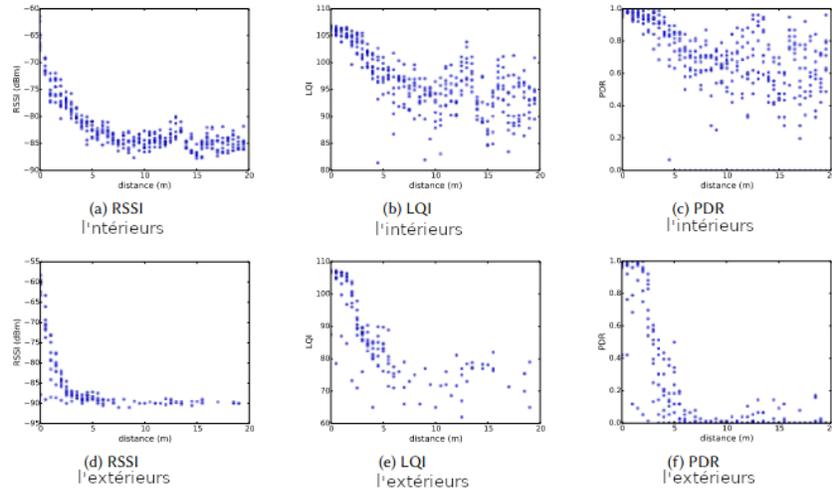


Fig. II.1 : Caractérisation de la fluctuation de la qualité des liaisons dans les environnements intérieurs et extérieurs [111]

Où $RSSI(d)$ est la perte de chemin à la distance d , $RSSI(d_0)$ est la perte de chemin mesurée à une distance de référence d_0 , n représente l'exposant de perte de chemin et $N(0, \sigma_b^2)$ est un bruit blanc-gaussien. L'exposant de perte de chemin est différent pour différents environnements. Selon la courbe, il est possible d'estimer l'exposant de perte de chemin pour ce cas, qui est de 0,89 pour l'environnement intérieur et de 1,12 pour l'environnement extérieur. Depuis les courbes c et f de la figure ci-dessus, nous observons que la qualité de la liaison en termes de PDR n'est pas strictement corrélée avec distance, bien que le PDR diminue généralement avec l'augmentation de la distance. Sur c, le PDR compris entre 10 et 14 m est supérieur à celui compris entre 5 et 10 m. Dans la plage de 14 à 15 m, le PDR diminue, puis après 16 m il augmente légèrement. Sur f, on observe le même phénomène (que les emplacements plus éloignés ont un PDR amélioré). De plus, dans l'expérience en extérieur, les liens sont presque déconnectés après 7 m où, comme dans l'environnement intérieur, même après 20 m, le taux de réussite des paquets était relativement élevé.

En observant la forme de l'intrigue du PDR, RSSI et LQI, on remarque que PDR est fortement corrélé avec RSSI et LQI. Ils ont presque la même tendance à la variation.

II.5. Prédiction de la qualité des liens de connectivité entre les WSN

L'établissement d'une connectivité RF (radiofréquence) adéquate est la condition fondamentale pour le bon fonctionnement de tout réseau sans fil. Dans un réseau mobile sans fil, il est difficile pour les applications et les protocoles de gérer les problèmes de connectivité, car les liaisons peuvent monter et descendre fréquemment. Dans ces scénarios, avoir connaissance du temps de connectivité RF restant du nœud peut éviter des transmissions de messages de contrôle/données inutiles.

II.5.1. Travaux connexes pour la prédiction de la connectivité

Le travail présenté dans [114] était l'un des précurseurs dans la prédiction de la connectivité entre les nœuds dans les environnements mobiles. Les auteurs ont présenté un modèle déterministe pour prédire la connectivité (voir l'équation ci-dessous). L'équation proposée peut donner le temps de connexion restant entre les nœuds mobiles. L'ensemble d'informations adopté est composé par la connaissance de la position (fournie par un GPS), de l'angle θ , de la vitesse v et de la plage de signal r des nœuds.

$$Dt = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2} \quad (\text{Equ.I.3})$$

Où Dt est la durée pendant laquelle deux nœuds mobiles resteront connectés

$$a = v_i \cos\theta_i - v_j \cos\theta_j$$

$$b = x_i - x_j$$

$$c = v_i \sin\theta_i - v_j \sin\theta_j$$

$$d = y_i - y_j$$

Dans [115], les auteurs ont développé une technique pour prédire les déconnexions de liens ciblant les applications collaboratives dans les MANET (pour "Mobile Ad Hoc Network". Un MANET est un type de réseau ad hoc qui peut changer d'emplacement et se configurer à la volée. Les MANETS étant mobiles, ils utilisent des connexions sans fil pour se connecter à divers réseaux. Il peut s'agir d'une connexion Wi-Fi standard ou d'un autre support, tel qu'une transmission cellulaire ou par satellite). Ils ont supposé que tous les nœuds avaient leur position fournie par un système GPS-free. Au moment du démarrage, tous les nœuds doivent être connectés pour créer un graphique (en passant, cela peut être considéré comme une limitation de cette

approche). Le poids de chaque arête représente la distance entre les nœuds voisins. La variation de cette distance est utilisée pour construire le graphe de prédiction. Avec un tel graphique, le nœud coordinateur peut prévoir quand un nœud se déconnectera et pourrait essayer de l'éviter.

Dans [116], la proposition des auteurs est de mesurer la durée des liens (ou la stabilité des liens) entre les nœuds. Le schéma de prédiction adopté combine la connaissance de la position avec des informations d'évanouissement de canal. Premièrement, ils utilisent leurs connaissances sur la mobilité des nœuds pour prédire la durée de vie de la liaison. Ensuite, ils combinent la durée de vie de la liaison avec les statistiques de canal d'évanouissement pour obtenir la durée de liaison. Pour calculer la durabilité de la liaison, ils supposent que tous les nœuds utilisent un GPS pour fournir leur emplacement et leur vitesse. Par conséquent, ils combinent la connaissance de la vitesse relative avec des statistiques de l'évanouissement du canal à la prévision de la durée de la liaison.

Un modèle permettant de mieux utiliser les ressources de communication d'un WLAN en prenant en compte l'historique de mobilité des utilisateurs est présenté dans [117]. Un tel historique est utilisé comme entrée d'un modèle de Markov, qui vise à prédire la mobilité des nœuds. L'approche nécessite des informations de positionnement, un appareil GPS est donc nécessaire. Ce schéma est basé sur la prédiction de localisation, car les nœuds et les routeurs ont leurs connaissances de positionnement, le modèle est construit en fonction de la probabilité de changer la localisation. Par exemple, si un nœud se déplace d'une zone à une autre, les routeurs peuvent prévoir le changement de localisation du nœud et anticiper la réservation de bande passante. Le principal inconvénient d'une telle approche est la nécessité d'un système de localisation.

Dans une approche différente, les auteurs de [118] utilisent la connaissance de la qualité des liens pour prédire la connectivité entre les nœuds. Ils ont utilisé une série chronologique pour modéliser les changements sur la qualité de la liaison au cours du temps pendant que les nœuds se déplacent. Cette approche nécessite de l'histoire. Dans un scénario prédéterminé, l'historique de la mobilité des utilisateurs a été utilisé comme paramètre d'entrée afin de construire le modèle. Pour cette raison, il n'était pas nécessaire d'utiliser du matériel supplémentaire pour obtenir des informations de positionnement.

L'approche MTCP présentée dans [119] représente les variations RSSI en utilisant un modèle de Markov à cinq états. Leur intention est d'utiliser MTCP pour améliorer le mécanisme d'interruption du protocole MAC, afin que les nœuds puissent économiser de l'énergie et également

améliorer la bande passante disponible. Les résultats obtenus sont très positifs compte tenu de la prévision d'une longueur d'avance.

Le modèle de *Markov* naissance-décès (Birth-Death ou tout court BD) présenté dans [120] a l'avantage d'éviter l'utilisation d'informations de positionnement. Au lieu de cela, il prédit la qualité des liaisons sans fil dans les réseaux Wifi en utilisant des informations de qualité de liaison au moyen du rapport signal sur bruit (SNR). La principale limitation de ce modèle est le fait qu'il ne représente pas correctement le mouvement des nœuds. Il ignore le principe d'inertie, qui stipule qu'un objet ne peut pas simplement changer de direction immédiatement. Un tel problème est exprimé par le modèle de Markov proposé, qui suggère qu'un nœud a la même probabilité d'augmenter ou de diminuer la force du signal, c'est-à-dire de maintenir ou de changer sa direction. Cependant, en fait, la probabilité que le nœud continue à aller dans la même direction est beaucoup plus élevée que de changer de direction. Ce problème a été surmonté par une proposition de [86] de naissance-mort orientée (OBD) [121], qui a été développé pour pallier le manque d'orientation du modèle BD [120].

En analysant tous les travaux connexes, on peut voir qu'ils utilisent des modèles de Markov ou des séries temporelles pour représenter le mécanisme de prédiction. De plus, on constate que toutes les solutions qui nécessitent un dispositif de localisation (comme un GPS) ont des restrictions à appliquer dans les scénarios intérieurs. En outre, des nœuds d'ancrage sont nécessaires avec les propositions sans GPS. Dans les deux cas, la reconnaissance de l'emplacement géographique est utilisée, augmentant la complexité de calcul de la solution et la rendant moins flexible.

En outre, certaines solutions ont également des restrictions strictes de modèle. Par exemple, le modèle dans [115] dépend du nombre de nœuds, puisque chaque nœud est représenté par un état dans la chaîne de *Markov*. Cela peut entraîner des problèmes d'évolutivité, car il peut être difficile de gérer un nombre élevé de nœuds. De plus, il ne convient pas aux applications avec un nombre variable de nœuds. L'inconvénient de [118] est qu'il ne traite pas des liaisons futures, c'est-à-dire que si un nouveau nœud voisin apparaît, le modèle ne peut pas prévoir la connectivité avec lui. Par conséquent, si le modèle de mobilité change, le modèle sera obsolète et une nouvelle formation hors ligne doit être effectuée.

Pour modèle ODB [121] a été développé pour pallier le manque d'orientation du modèle BD [120]. Pour mieux comprendre ce problème, quelques mesures présentés dans [86] contenant la variation du SNR pendant qu'un robot au sol se déplace dans une pièce, envoyant des messages à

un nœud de capteur statique situé dans le coin de la pièce. Ces résultats sont présentés sur la figure (II.2). La ligne avec des perturbations (couleur plus claire) représente le SNR brut et la plus stable (couleur plus foncée) représente le SNR filtré avec un filtre de Kalman. En fait, la courbe la plus stable représente le modèle de mobilité du robot. Lorsque le robot s'approche du nœud de capteur, la force du signal RF atteint son maximum. De même, Lorsque le robot s'éloigne du nœud de capteur, la force du signal RF diminue jusqu'à ce qu'il n'y ait plus d'alimentation en termes de liaison RF. L'approche et le départ sont des modèles qui existent dans tous les types de mobilité.

Tout comme le modèle BD, OBD utilise également un modèle de chaîne de Markov pour prédire la qualité du lien dans le futur. Les modèles de Markov ont l'avantage de ne pas avoir besoin de conserver l'historique en mémoire, ce qui est un bon point pour les périphériques à ressources limitées comme un nœud de capteur. En fait, l'histoire est déjà intégrée dans la probabilité de transitions d'états du modèle de Markov. Ces probabilités sont stockées dans une matrice de taille fixe.

La principale différence entre O-BD et BD repose sur le fait que le premier prend en compte l'orientation/direction du mouvement des nœuds. Cela ne signifie pas qu'un nœud ne peut pas changer de direction. Ce qui veut dire que cela a une très faible probabilité de se produire. Pour représenter un tel problème, tous les états intermédiaires de l'automate sont dupliqués, comme le montre la figure (II.3). Les états en haut représentent les nœuds dans une direction, et l'état en bas représentent les nœuds dans la direction opposée.

Les états à l'extrême ne sont pas dupliqués car il n'a de sens pour eux d'aller que dans une seule direction. Les deux caractéristiques d'approche et de départ, visibles sur la figure (II.2), sont présentées dans O-BD.

Les connaissances fournies par le modèle sont utiles pour permettre de prévoir l'état suivant. Par conséquent, les auteurs de [86] ont utilisé la théorie du modèle de Markov discret, où chaque transition se produit en une seconde. La chaîne de Markov discrète pourrait être décrite comme un triple $\langle S, T, \pi \rangle$. S représente l'ensemble des états, comme illustré sur la figure (II.3), et T est la matrice de transition qui représente la probabilité de passer d'un certain état i à l'état j . Il est lié à chaque scénario possible (modèle de mobilité) et peut être obtenu par simulation ou expérimentation pratique. Enfin, π est la distribution de probabilité initiale de l'ensemble S . Pour choisir l'état suivant, on utilise la probabilité d'état vectoriel, comme le montre l'Equ.I.4. T^n est la matrice de transition alimentée par n (nombre de pas en avant). Après ce calcul, il est possible

d'obtenir le nouveau vecteur de probabilité π^n et enfin de choisir l'état suivant en utilisant Equ.1.5. Comme pour tous les modèles basés sur une histoire, la chaîne de Markov discrète est affectée par la modification du modèle de mobilité.

$$\pi^n = \pi \times T^n \tag{Equ.I.4}$$

$$s^n = \text{Max}[s_1^n, s_2^n, \dots, s_k^n,] \tag{Equ.I.5}$$

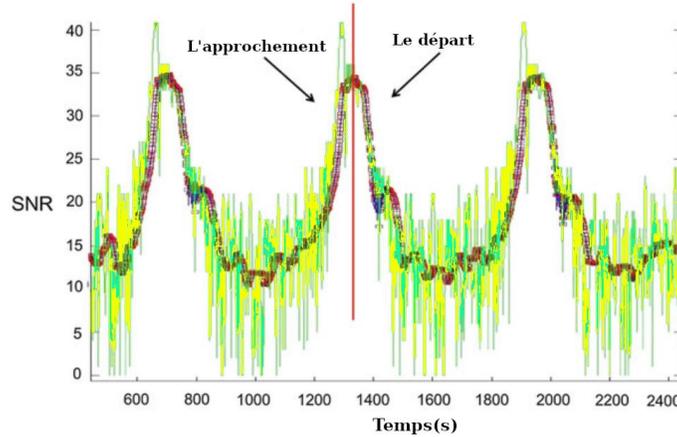


Fig. II.2 : Le SNR en fonction de la situation d'approche et de départ au fil du temps [10]

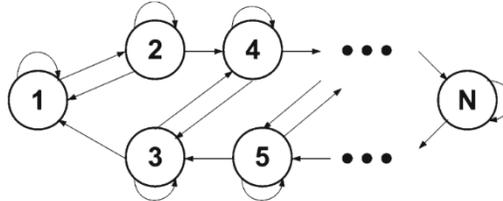


Fig. II.3 : Automate orienté naissance-mort (O-BD) [10]

II.5.2. GMLA pour la prédiction de la qualité des liaisons dans les WSNs

Une approche de [86] dite d'apprentissage automatique génétique pour la prédiction de la qualité des liens, ou simplement GMLA, qui est une solution pour prévoir le temps de connectivité RF restant dans les environnements mobiles. Contrairement à tous les travaux connexes, GMLA permet de construire des connaissances en connectivité pour estimer la durée de la liaison RF sans avoir besoin d'une phase préalable. Cela permet d'appliquer GMLA dans des environnements et des modèles de mobilité inconnus. Sa structure combine un système de classification avec un modèle de chaîne de Markov de qualité de liaison RF. Comme les paramètres du modèle de Markov sont découverts à la volée, il n'est pas nécessaire d'avoir un historique antérieur pour alimenter le modèle de Markov. Des résultats de simulation obtenus dans [86] montrent que

GMLA est une solution très appropriée, car elle surpasse les approches qui utilisent des systèmes de positionnement géographique (GPS) et également des approches qui utilisent la prédiction de qualité de liaison, comme BD et MTCP. GMLA est suffisamment générique pour être appliqué à n'importe quelle couche de la pile de protocoles de communication, en particulier dans les couches liaison et réseau.

Récemment, il est apparu un besoin éminent d'utiliser des réseaux de capteurs sans fil (WSN) dans des scénarios traitant de la mobilité. Un scénario typique pourrait être des robots se déplaçant dans une usine tout en communiquant avec d'autres robots et également avec des nœuds fixes pour effectuer un travail collaboratif [122]. Un autre exemple est la collaboration entre les WSN et les véhicules aériens sans pilote (UAV), qui visent la détection de l'emplacement et le suivi des cibles [123-126].

Cependant, déployer de tels réseaux mobiles sans fil n'est pas une tâche facile [123]. L'un des problèmes qui présente un intérêt particulier concerne la connectivité radiofréquence (RF), car l'établissement d'une connectivité RF appropriée est la condition de base pour le bon fonctionnement de tout réseau sans fil. Dans les applications mobiles, les protocoles réseau sont mis au défi de gérer les problèmes de connectivité RF, car les liaisons peuvent monter et descendre fréquemment. Dans ces scénarios, avoir connaissance du temps de connectivité restant du nœud peut éviter des transmissions de messages de contrôle ou données inutiles. Cette propriété est particulièrement utile pour les protocoles de couche réseau et liaison.

La prise de conscience des variations de connectivité dans les environnements mobiles est généralement gérée en utilisant des stratégies de formation basées sur la localisation ou des données. Alors que les stratégies basées sur la localisation nécessitent une sorte de système de localisation, les stratégies de formation de données reposent sur les données recueillies à partir de la carte réseau [127, 128]. Les deux stratégies garantissent la construction d'une sorte de connaissance de la connectivité.

Contrairement à tous les travaux connexes, une solution présentée permet de construire une connaissance de la connectivité afin d'estimer la durée de la liaison sans avoir besoin d'une phase de pré-exécution (conception). Il peut effectuer la prédiction de connectivité alors que le système est déjà en cours d'exécution (ce mécanisme est appelé « on-the-fly ou bien à la volée »). La solution est appelée approche d'apprentissage automatique génétique pour la prédiction de la qualité des liens (Genetic Machine Learning Approach for Link Quality Prediction), ou

simplement GMLA. Pour pouvoir être performant à la volée, GMLA utilise une approche évolutive qui apprend le modèle de mobilité le long des premières secondes de l'exécution de l'algorithme. GMLA est suffisamment générique pour être appliqué à n'importe quelle couche de la pile de protocoles de communication, en particulier dans les couches liaison et réseau.

L'approche d'apprentissage automatique génétique pour la prédiction de la qualité des liens (GMLA) est basée sur des systèmes de classification [129], une technique d'apprentissage automatique basée sur des algorithmes génétiques (AG ou GA pour Genetic Algorithms) qui est capable d'apprendre des règles syntaxiquement simples. Il permet de découvrir lors de l'exécution les paramètres du modèle de Markov qui représentent la qualité RF. La figure (II.4) ci-dessous présente un aperçu de l'approche proposée, ou de celle n'importe quel système de classification similaire.

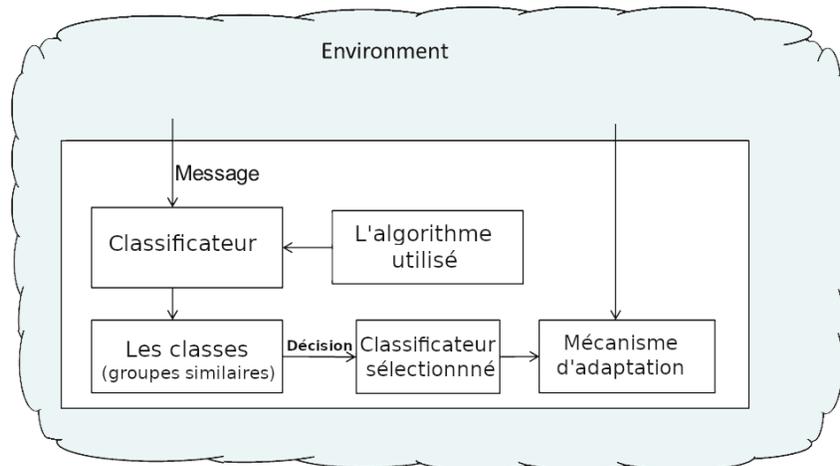


Fig. II.4 : Schéma du système de classification

Un message reçu de l'environnement peut activer un ou plusieurs classificateurs. Au fur et à mesure que les classificateurs sont sélectionnés, ils exécutent leurs règles. Ensuite, les classificateurs sélectionnés sont récompensés (adaptés) en fonction de leur performance. Les GA considèrent un ensemble de classificateurs pour un problème d'optimisation. Dans ce cas, il y a des individus (classificateurs) qui présentent leurs génotypes, qui sont généralement un ensemble de bits ou de caractères. Cet ensemble est évolué par l'AG après un nombre de consultations redéterminé. A chaque génération de réponses, un nouvel ensemble de créatures artificielles (classificateurs) est généré. Les réponses sont basées sur des fragments d'individus antérieurs les plus adaptés.

L'objectif principal d'un GA est sa robustesse. Si le système est plus robuste, il nécessite un plus petit nombre d'interventions ou de redéfinitions. De plus, il atteindra des niveaux d'adaptation plus élevés et pourra mieux fonctionner et durer plus longtemps. La principale différence entre une approche GA classique et un système de classificateurs est que ce dernier fait évoluer sa population juste après quelques consultations avec l'ensemble de classificateurs. Ainsi, il effectuera le processus d'évolution pendant son exécution.

L'approche d'apprentissage machine génétique proposée (GMLA) est une extension du modèle OBD [121], car elle utilise également un modèle de chaîne de Markov pour prédire la qualité des liens dans le futur.

Comme mentionné, la matrice de transition T utilisée par les modèles de Markov dépend des caractéristiques de l'environnement, comme le modèle de mobilité. Pour surmonter ce problème et permettre à la même pile de protocoles d'être utilisée dans des conditions de mobilité inconnues, il a développé par [86] un algorithme évolutif capable de trouver le meilleur T à la volée. Pour π , ils ont supposé que les états soient équiprobables (chaque état a la probabilité $1/m$, où m est le nombre d'états du modèle).

Un nœud peut commencer à communiquer à n'importe quelle plage de signaux. Le temps nécessaire pour entraîner le modèle et trouver le meilleur T varie en fonction de la quantité d'échantillons de qualité de liaison disponibles. Bien sûr, cela dépend de chaque scénario d'application. Au fur et à mesure que les échantillons de qualité de liaison arrivent avec une fréquence plus élevée, plus le modèle peut être construit rapidement. L'inverse est également vrai. Cependant, cette phase d'apprentissage n'est pas toujours nécessaire. Par exemple, les nœuds peuvent recevoir T d'autres nœuds qui ont déjà appris, ou même de l'environnement lui-même. Chaque chromosome dans GMLA représente une matrice de transition et chaque gène représente une ligne, qui est l'état dans l'automate. La figure (II.5) ci-dessous représente l'idée d'un chromosome.

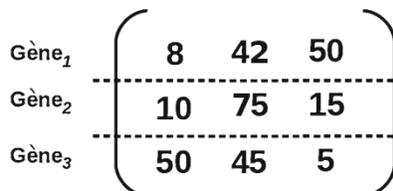


Fig. II.5 : Le chromosome en tant que matrice et les lignes comme gènes

Pour chaque gène (une ligne), il contient les valeurs des probabilités de transitions pour passer de l'état i à j . Chaque gène doit garantir le rôle de base du modèle de Markov, qui est la somme des probabilités de transitions à partir d'un état qui doit totaliser 100%. Pour garantir cette propriété, chaque gène exécute un algorithme dont ; par exemple, si le gène a trois transitions, la première transition sélectionnée aléatoirement comme un nombre $N1$ entre 0 et 100, puis la deuxième transition sélectionne aléatoirement un nombre $N2$ entre 0 et $(100 - N1)$ et le dernier prend le reste, $N3 = 100 - (N1 + N2)$. Au final, la somme de $N1 + N2 + N3$ est de 100%. De cette manière, chaque chromosome est créé et représente le modèle de l'automate.

Lors du démarrage, chaque individu est généré aléatoirement, puis il évolue selon les phases illustrées sur la figure (II.6) ci-après. Après avoir échantillonné la qualité de liaison, les individus mappent la valeur de qualité de liaison avec l'état actuel. L'étape suivante, ils devinent l'état suivant pour N transitions à venir en calculant les équations. 2 et 3. Après N étapes supplémentaires, tous les individus sont évalués. La qualité de la liaison est échantillonnée et mappée à nouveau, puis elle est comparée à l'état supposé. Tous les individus qui ont bien deviné sont récompensés. Ce processus se poursuit jusqu'à ce que le nombre de requêtes se termine.

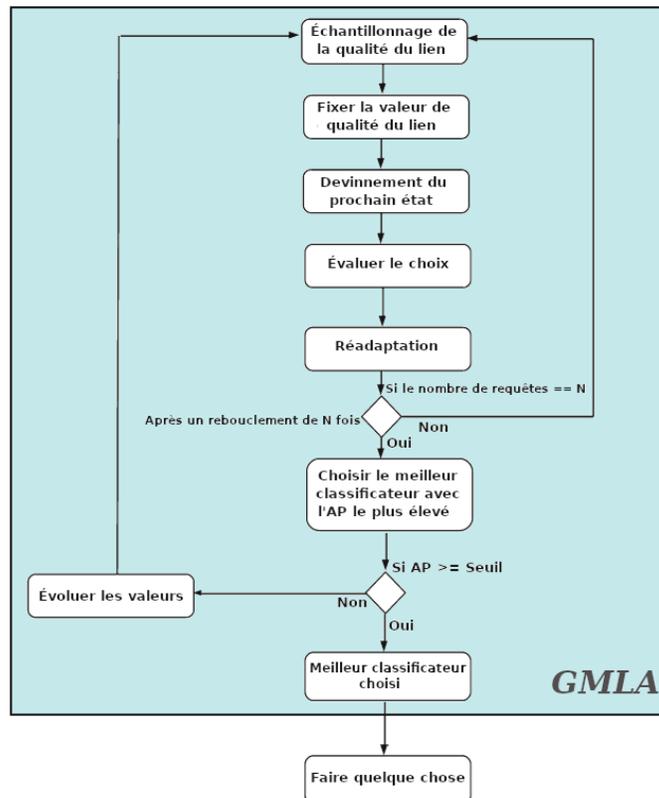


Fig. II.6 : L'approche d'apprentissage automatique génétique pour la prédiction de la qualité des liens

Après un certain nombre de requêtes, l'étape suivante consiste à choisir le meilleur classificateur, en appliquant la fonction d'ajustement (Fitness Function) dont $AP = \sum P_t Q$. Le pourcentage de précision (AP) est le nombre de suppositions correctes P_t divisé par le nombre de requêtes Q . Le classificateur avec AP le plus élevé est choisi. Si le meilleur AP du classificateur est supérieur ou égal à un seuil déterminé, le processus GMLA se termine. Cependant, si le meilleur AP du classificateur est inférieur au seuil déterminé, toute la population évolue. Le processus d'évolution est composé par croisement et mutation. Le croisement est appliqué aux deux meilleurs individus (ceux avec le PA le plus élevé) et leurs gènes (états) sont croisés. Par exemple, dans un modèle à 3 états, l'état 1 est choisi parmi l'individu A et les états 2 et 3 sont choisis parmi l'individu B, générant deux nouveaux individus.

Après le croisement, tous les individus qui ont une valeur AP inférieure à un certain seuil sont mutés. La mutation est appliquée dans un état choisi au hasard. Les transitions sont également choisies au hasard en appliquant l'algorithme précédent (de N1, N2 et N3 comme exemple), de sorte que de nouvelles probabilités sont obtenues. Par conséquent, les deux pires individus sont remplacés par de nouveaux individus. Ces étapes sont appelées phase d'apprentissage. Ils sont répétés jusqu'à ce qu'un individu avec AP au-dessus du seuil soit obtenu. Enfin, le meilleur individu est choisi (celui qui a l'AP le plus élevé).

II.6. Analyse des technologies de transmission sans fil (systèmes multirobots mobiles)

De nos jours, l'un des principaux défis de la communauté robotique est de sortir les robots des laboratoires et de les utiliser dans des applications réelles. Même si d'importantes ressources ont été consacrées à cette direction, le défi demeure. De plus, lorsque l'on travaille avec des systèmes multirobots, des complications encore plus grandes surviennent [130].

La communication est un problème clé lorsqu'il s'agit de systèmes multirobots. Indépendamment des solutions centralisées ou décentralisées, il existe toujours un fort besoin de communiquer avec et entre les robots et le poste de contrôle. La plupart des systèmes multirobots utilisent les communications sans fil pour éviter les restrictions sur les mouvements des agents [131]. Par conséquent, la robustesse des communications en termes de bande passante, de portée ou de latence s'avère être un aspect crucial lors de l'évaluation des performances globales du système. La forme de communication la plus simple est un système point à point, dans lequel les agents des systèmes multirobots envoient des informations directement à leur destinataire. Cependant, cette approche n'est adaptée que lorsque le nombre de composants du système multi-

robot est faible, car lorsque l'on travaille avec de nombreux robots, la saturation des canaux peut être facilement atteinte et des retards ou d'autres problèmes surviennent. Des approches plus complexes, telles que les schémas un-à-plusieurs ou plusieurs-à-plusieurs, sont également largement utilisées. De plus, un schéma très courant est basé sur l'échange d'informations entre des robots proches les uns des autres, bien qu'ils nécessitent un processus supplémentaire pour trouver quels autres robots sont proches, et parfois les combiner avec des règles simples.

L'application du système multirobot définit les deux principales exigences de communication, qui sont la portée et la bande passante, et par conséquent, la solution technique possible à utiliser. Actuellement, plusieurs technologies sans fil existantes sont disponibles sur le marché à un coût abordable. Quelques exemples sont le Wi-Fi (normes IEEE 802.11a/b/g/n/ac), l'interopérabilité mondiale des réseaux d'accès micro-ondes connus sous le nom de WiMax (IEEE 802.16), Sigfox, la 3G, la 4G LTE ou la 5G, ZigBee et LoRa [132].

Compte tenu des applications qui nécessitent de couvrir de grandes zones (près de 1 Km), les technologies WiMax, Sigfox et les réseaux téléphonique sont a priori les plus adéquates. Néanmoins, d'un point de vue économique, le coût du matériel, des licences ou du prestataire de services les rend inadaptés. Les solutions Wi-Fi sont cependant bien plus intéressantes à cet égard [132].

De plus, la plupart des ordinateurs et appareils utilisent le Wi-Fi, il n'est donc pas nécessaire de recourir à des appareils de pont supplémentaires. De plus, le problème de la portée peut être résolu en ajoutant des répéteurs ou des prolongateurs de portée. Habituellement, les systèmes de communication pour système multirobot ont la majeure partie de leur puissance à un emplacement fixe ou semi-mobile à proximité de la station de base, en raison de la consommation d'énergie et de la taille de l'antenne de la station de contrôle, cela pose rarement un problème, alors qu'ils peuvent être graves. Restreint dans les robots. Cela signifie que la taille des antennes embarquées dépend de la taille des robots et de l'environnement dans lequel ils évolueront. De plus, le poids pourrait poser des problèmes, surtout lorsqu'il s'agit de véhicules aériens avec une charge utile limitée. La bande passante peut être un problème crucial lorsque l'on travaille avec système multirobot, le nombre de nœuds, de sujets et d'échanges d'informations peut croître de façon exponentielle lorsque la taille du système multirobot augmente. Néanmoins, dans la plupart des applications, chaque robot s'appuie sur un ordinateur embarqué. Cet ordinateur héberge généralement les nœuds publiant la plupart des données volumineuses, qui sont généralement des

données brutes de capteurs tels que des lasers, des caméras, des odomètres, des mesures inertielles, etc [133]. De plus, le même ordinateur héberge également la plupart des nœuds qui utilisent ces informations (différents niveaux de contrôle, navigation, etc). Cela signifie qu'une bande passante plus élevée est requise à l'intérieur du robot, alors qu'entre les robots, les communications peuvent être moins intensives, envoyant uniquement des commandes de haut niveau et des rapports de télémétrie ou d'état de mission réduits. Cela conduit à la nécessité de concevoir une bonne architecture logicielle pour maintenir l'échange de données important à l'intérieur du robot, réduire la quantité de communications entre les robots et donc nécessiter moins de bande passante des systèmes de communication [134].

La transmission par radiofréquence (RF) est un aspect crucial dans la numérisation des robots collaboratifs décentralisés (cobots) dans des environnements hostiles. Les cobots sont des robots très polyvalents qui peuvent être utilisés dans divers environnements industriels et manufacturiers, allant des salles blanches aux environnements dangereux et éloignés. La capacité de transmettre des données et des signaux de contrôle sans fil via la transmission RF est essentielle au fonctionnement fluide de ces robots. [135]. L'objectif est d'échanger sur les différents moyens de transmission RF et d'identifier la meilleure solution pour les cobots en environnement hostile.

Les cobots peuvent faire partie intégrante de l'industrie et de tous les lieux dans lesquels les humains existent et travaillent, grâce à leur capacité à travailler aux côtés des travailleurs humains, renforçant ainsi leurs capacités et améliorant la productivité globale. Ces robots sont conçus pour fonctionner dans un environnement collaboratif, offrant un haut niveau de flexibilité et d'adaptabilité aux exigences changeantes de la production. Dans les environnements hostiles, tels que ceux présentant des niveaux élevés d'interférences, de chaleur et de poussière, les cobots doivent être équipés de technologies de transmission RF robustes et fiables.

II.6.1. Technologies de transmission pour robots mobiles

Technologies de transmission RF actuelles pour des robots collaboratif mobiles sont généralement le Wi-Fi, Zigbee, Lora et le Bluetooth [136-138]. Dans les sections suivantes, nous discuterons de ces technologies brièvement et analyserons leur adéquation aux cobots dans des environnements hostiles (Tableau II.1).

II.6.1.1. Wi-Fi

Le Wi-Fi est une technologie de transmission RF populaire qui utilise les bandes de fréquences 2,4 GHz et 5 GHz pour la communication. Le Wi-Fi est largement utilisé dans les environnements

résidentiels et commerciaux et constitue un moyen courant de transmission de données et de signaux de contrôle dans les cobots. Cependant, le Wi-Fi présente certaines limites dans les environnements hostiles en raison de sa sensibilité aux interférences provenant d'autres appareils utilisant la même bande de fréquences. Cela peut entraîner une transmission de données lente et une mauvaise connectivité, ce qui le rend inadapté aux cobots dans des environnements industriels difficiles. [139].

II.6.1.2. Zigbee

Zigbee est une technologie de transmission RF à faible consommation et à faible débit qui utilise la bande de fréquences de 2,4 GHz. Zigbee est conçu pour fournir une communication sans fil fiable et sécurisée dans les réseaux maillés, ce qui en fait un choix populaire pour les systèmes d'automatisation industrielle. Cependant, Zigbee a une portée limitée, ce qui le rend inadapté aux cobots dans les grands environnements industriels tentaculaires [140].

II.6.1.3. Bluetooth

Bluetooth est une technologie de transmission RF de faible puissance qui utilise la bande de fréquences de 2,4 GHz. Le Bluetooth est largement utilisé pour la communication sans fil entre des appareils, tels que les smartphones et les ordinateurs portables, et est également utilisé dans les cobots pour la transmission de données et de signaux de contrôle. Bluetooth a une portée limitée, ce qui le rend inadapté aux grands environnements industriels. De plus, le Bluetooth est sensible aux interférences provenant d'autres appareils utilisant la même bande de fréquences, ce qui le rend peu fiable dans des environnements hostiles [141].

II.6.1.4. Lora/LoraWAN

LoRa (abréviation de « Long Range ») est une technologie de communication sans fil développée en 2012 par Semtech Corporation qui permet une communication longue portée et à faible consommation entre les appareils. LoRa fonctionne dans le spectre radio sans licence et utilise des techniques de modulation à spectre étalé pour réaliser une communication à longue portée tout en consommant très peu d'énergie. LoRa est souvent utilisé dans des applications telles que les villes intelligentes, l'automatisation industrielle et l'agriculture.

LoRaWAN (abréviation de « Long Range Wide Area Network ») est un protocole réseau construit sur la technologie LoRa. LoRaWAN fournit un protocole de communication standard permettant aux appareils IoT de se connecter à un réseau LoRa. Il offre une communication sécurisée et bidirectionnelle entre les appareils IoT et les passerelles, et permet une évolutivité et

une flexibilité dans le déploiement du réseau. LoRaWAN est un standard ouvert, ce qui signifie qu'il peut être mis en œuvre par n'importe qui et n'est pas la propriété de Semtech Corporation. [142].

La principale différence entre LoRa et LoRaWAN est que LoRa est une technologie de communication sans fil qui permet une communication longue portée et à faible consommation entre les appareils, tandis que LoRaWAN est un protocole de réseau construit sur la technologie LoRa qui fournit un protocole de communication standard pour les appareils IoT pour connecter à un réseau LoRa. En d'autres termes, LoRa est la technologie de couche physique, tandis que LoRaWAN est le protocole de couche réseau qui permet aux appareils de communiquer entre eux et avec l'infrastructure réseau. Une autre différence importante est que LoRaWAN fournit des fonctionnalités telles que la sécurité, la gestion du réseau et l'authentification des appareils, qui sont nécessaires aux déploiements IoT à grande échelle. LoRa, en revanche, est une technologie de couche physique qui ne fournit pas ces fonctionnalités à elle seule [138, 143, 144].

LoRaWAN est particulièrement adapté aux cobots dans des environnements hostiles, car il résiste aux interférences et peut transmettre des données sur de longues distances.

Tab. II.1 : Comparaison générale

CRITERE	WI-FI	BLUETOOTH	ZIGBEE	LORA
BANDE DE FREQUENCE	2,4 GHz, 5 GHz	2,4 GHz (2400-2483,5MHz)	2,4 GHz (Global), 868 MHz (Europe), 915 MHz (Amérique du Nord), 915 MHz (Chine)	868 MHz (Europe), 915 MHz (Amérique du Nord), 433 MHz (Asie)
CANAUX	Jusqu'à 14 (2,4 GHz), Variables (5 GHz)	79 (2,4 GHz), 40 (LE)	16 (2,4 GHz), 10 (868 MHz), 10 (915 MHz)	Variables, dépend du pays (par exemple, 64 canaux en LoRaWAN)
TYPE DE MODULATION	OFDM, DSSS, FHSS	GFSK, $\pi/4$ DQPSK, 8DPSK, FHSS	BPSK, QPSK	CSS (Étalement de spectre par étalement de fréquence), LoRa, FSK
DEBIT DE DONNEES	600 Mbps - 7 Gbps	1 Mbps - 3 Mbps	20-250 kbps	0.3 kbps - 50 kbps (selon le facteur d'étalement)
PUISSANCE DE TRANSMISSION	Généralement plus élevée	Modérée	Faible	Faible à modérée
CONSOMMATION D'ENERGIE	Relativement élevée	Faible à modérée	Très faible	Très faible

Chapitre II : Connectivité des systèmes multirobot et réseaux de capteurs sans-fils (WSN)

SENSIBILITE DU RECEPTEUR	Modérée à élevée	Modérée	Élevée	Élevée
TOPOLOGIE DE COMMUNICATION	Point-à-Point, Point-à-Multipoint, Maillage	Point-à-Point, Point-à-Multipoint, Maillage	Point-à-Point, Point-à-Multipoint, Maillage	Point-à-Point, Point-à-Multipoint, Maillage
TOPOLOGIE RESEAU	Infrastructure, Ad-hoc, Maillage	Piconet/Scattern et	Maillage	Étoile, Point-à-Point, Maillage
PORTÉE	Courte à moyenne portée (30-100 m)	Portée courte (10-20 m, peut-être 100 m avec BLE 5.x)	Courte à moyenne portée (10-75 m)	Longue portée (Plusieurs kilomètres en milieu rural, jusqu'à quelques centaines de mètres en milieu urbain)
IMMUNITE CONTRE LES OBSTACLES (SHADOWING)	Modérée à faible	Modérée à élevée	Élevée	Élevée
SECURITE	WPA3, WPA2, WEP, LEAP, Ouvert	Cryptage AES, Pairing (couplage)	AES-128, AES-256, Sécurité de la couche liaison	AES-128, échange de clés sécurisé
APPLICATIONS	Accès à Internet, Diffusion vidéo, Applications IoT, Domotique, Réseaux d'entreprise	Réseaux personnels (PAN), Wearables, Diffusion audio	Domotique, Automatisation industrielle, Santé, Compteurs intelligents, Applications IoT	IoT, Villes intelligentes, Agriculture, Suivi d'actifs, Surveillance de l'environnement, Gestion des catastrophes
UTILISATION DANS UN CLUSTER DE ROBOTS MOBILES	Oui	Oui	Oui	Oui
REFERENCES	[145-151]	[152-158]	[159-165]	[166-172]

II.6.2. Méthodologie Expérimentale

Dans le domaine en constante évolution de la robotique mobile, l'établissement d'une communication fiable et efficace entre les robots joue un rôle crucial, notamment dans des environnements hostiles où la collaboration entre ces entités est essentielle pour accomplir des missions complexes. Cette section se penche sur une étude expérimentale comparative de plusieurs technologies de transmission sans fil, chacune adaptée à des exigences spécifiques, dans le contexte de robots mobiles collaborant au sein de milieux difficiles.

Nous explorons les performances de différentes technologies, à savoir LoRa, Wifi, Bluetooth et Zigbee, dans des scénarios où la fiabilité, la portée, et la résilience face aux interférences sont

des facteurs déterminants. Notre démarche expérimentale visait à évaluer de manière approfondie les performances des technologies de transmission sans fil dans des conditions environnementales hostiles. Nous avons choisi quatre technologies couramment utilisées dans le domaine de la robotique mobile collaborative : Wifi, Bluetooth, LoRa, et Zigbee, représentées respectivement par les modules Edimax EW-7611ULB pour le WIFI et Bluetooth, Waveshare SX1262 LoRa HAT pour LoRa, et XBee S2C TH pour le ZigBee (Fig .II.7).

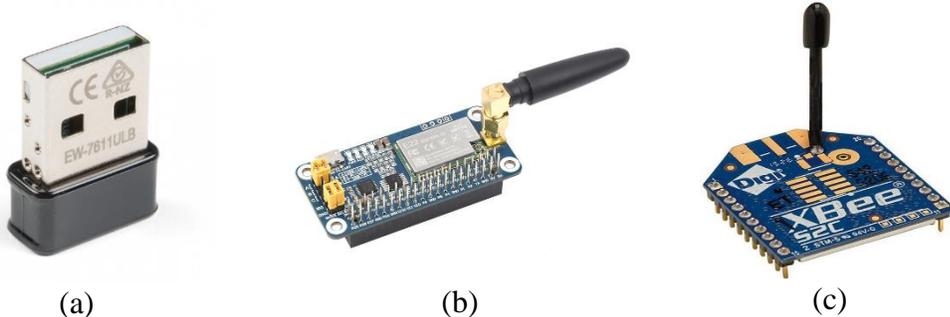


Fig. II.7 : les modules de transmissions utilisés, (a) clé Wifi+ Bluetooth Edimax EW-7611ULB, (b) le module de Raspberry Pi Waveshare SX1262 LoRa HAT, (c) le module de transmission par ZigBee XBee S2C TH

Cette comparaison approfondie vise à offrir des perspectives précieuses pour guider le choix de la technologie de transmission en fonction des exigences spécifiques de la mission, ouvrant ainsi la voie à des systèmes de robots collaboratifs performants et adaptés à des environnements souvent imprévisibles et hostiles.

II.6.2.1. Configuration Expérimentale

Nous avons mis en place un dispositif expérimental basé sur la communication point à point pour chaque technologie, simulant la communication entre des robots mobiles travaillant ensemble dans un environnement complexe (Figure II.8). Les distances entre les points ont été systématiquement variées pour couvrir une plage significative, allant de courtes distances à des valeurs plus grandes, reflétant des scénarios réalistes. De plus, chaque robot est intégré à un Raspberry Pi pour le traitement et le contrôle des données. Les robots utilisent des configurations et des algorithmes pour extraire des informations clés, qui sont ensuite stockées dans des bases de données locales au sein de chaque robot pour une analyse ultérieure. Les robots utilisent un processus d'odométrie pour une navigation précise, tandis que la technologie du système de positionnement global (GPS) est utilisée pour des expériences à grand déplacement, en particulier dans le cas de la communication LoRa. Obstacles fixes et dynamiques présents entre les robots, assurant l'environnement hostile de la transmission des données. Un ordinateur central est utilisé

pour collecter les données de leurs bases de données locales. Les signaux reçus de chaque robot sont capturés et utilisés pour une analyse statistique, permettant une évaluation complète des performances des différentes technologies de transmission dans des conditions environnementales hostiles. Dans l'ensemble, cette configuration expérimentale fournit un cadre scientifiquement rigoureux pour évaluer et comparer les performances de différentes technologies de transmission, apportant ainsi des informations précieuses dans le domaine de la robotique mobile dans des environnements hostiles.

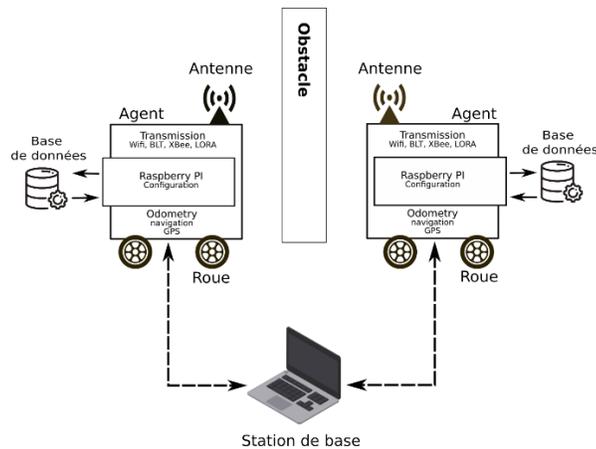


Fig. II.8 : Illustration de la configuration expérimentale

II.6.2.2. Mesures de Performance

- Calcul du RSSI en Fonction de la Distance : À chaque distance prédéfinie, nous avons mesuré le RSSI 'Received Signal Strength Indication' pour chaque technologie. Cette métrique offre un aperçu de l'intensité du signal reçu, cruciale pour évaluer la robustesse de la communication dans des environnements perturbés.
- Calcul du SNR en Fonction de la Distance : De manière similaire, nous avons enregistré le Signal-to-Noise Ratio (SNR) à chaque distance. Le SNR fournit une indication de la qualité du signal par rapport au bruit ambiant, un paramètre essentiel pour anticiper les interférences potentielles.
- Débit en Fonction de la Distance : Pour évaluer la vitesse de transmission des données, nous avons mesuré le débit en fonction de la distance pour chaque technologie. Cela permet de comprendre comment la performance des transmissions varie dans des environnements dynamiques et hostiles.

II.6.3. Résultats et Discussion

Nos expérimentations ont été menées dans un environnement réaliste, intégrant des obstacles tels que murs, eau, fer, et personnes. Ces conditions reproduisent des situations hostiles telles que des zones de mines ou des bâtiments impactés par des séismes, où la communication fiable entre robots est cruciale malgré les perturbations extérieures.

Cette approche expérimentale rigoureuse vise à fournir des données fiables et détaillées, offrant ainsi une base solide pour le choix optimal de la technologie de transmission dans des contextes réels et hostiles. Les résultats obtenus (Figures de II.9 à II.20) seront analysés de manière approfondie pour dégager des conclusions significatives et orientées vers des applications concrètes dans le domaine de la robotique mobile collaborative.



Fig. II.9 : RSSI en fonction de la distance, la technologie ZigBee

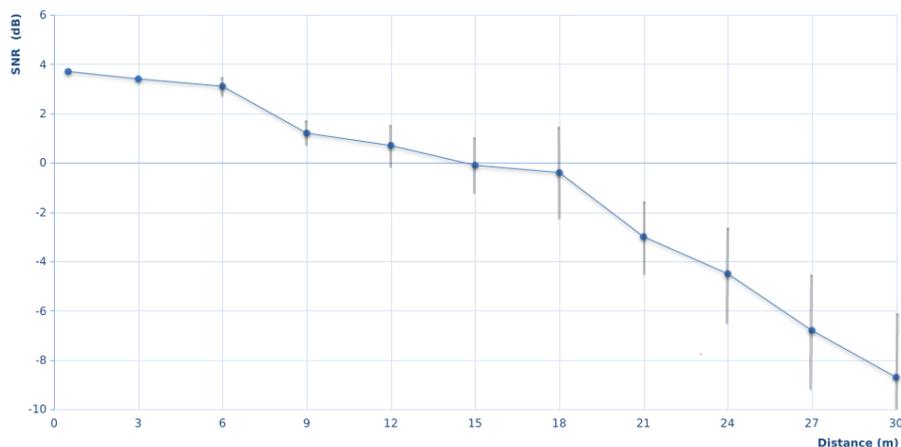


Fig. II.10 : SNR en fonction de la distance, la technologie ZigBee

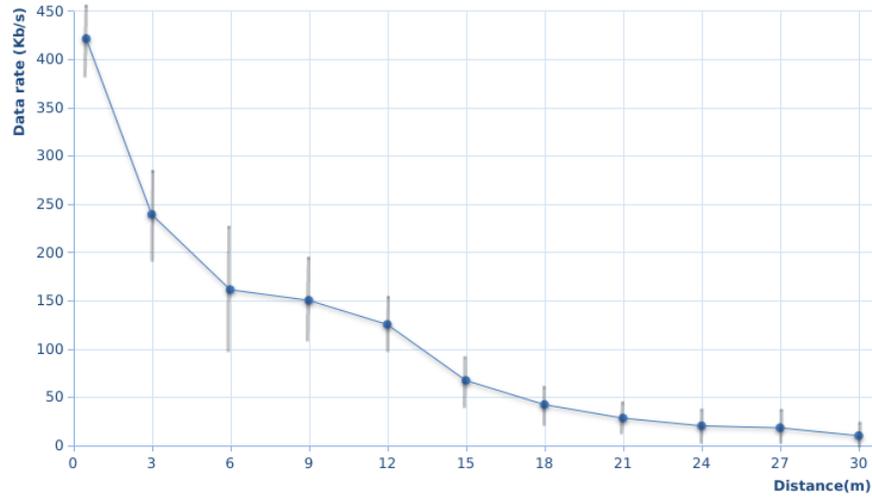


Fig. II.11 : Débit en fonction de la distance, la technologie ZigBee

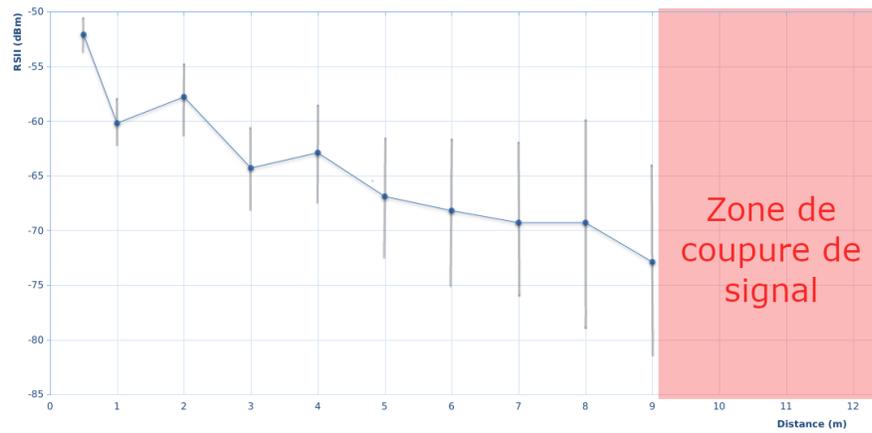


Fig. II.12 : RSSI en fonction de la distance, la technologie Bluetooth

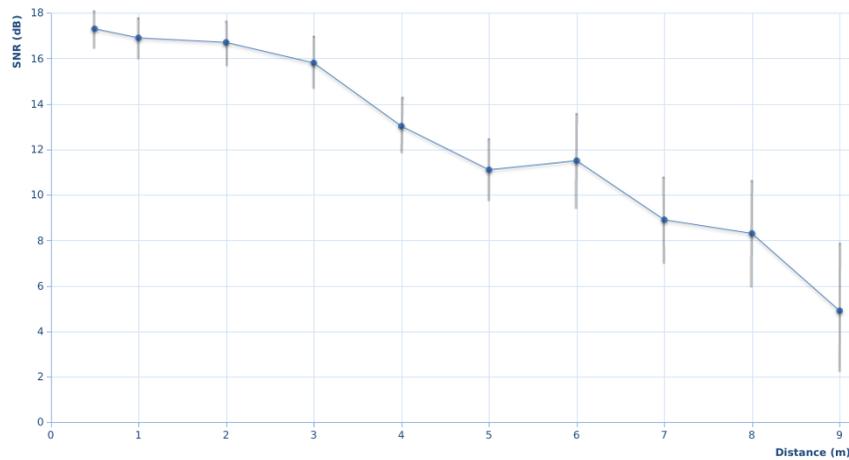


Fig. II.13 : SNR en fonction de la distance, la technologie Bluetooth

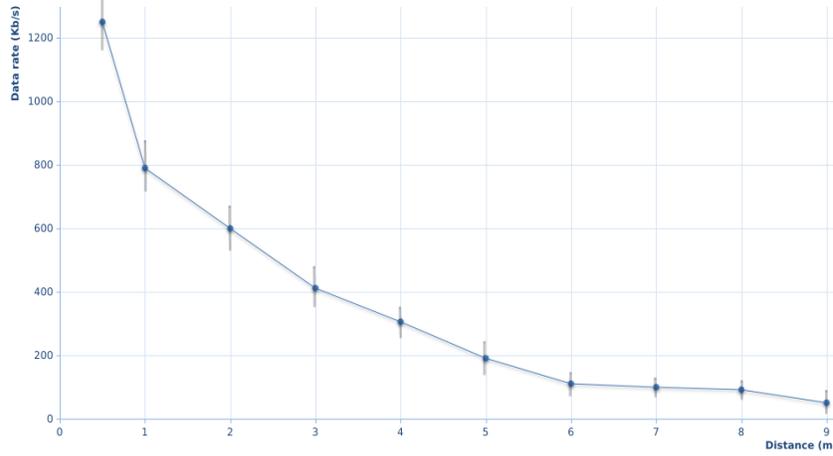


Fig. II.14 : Débit en fonction de la distance, la technologie Bluetooth

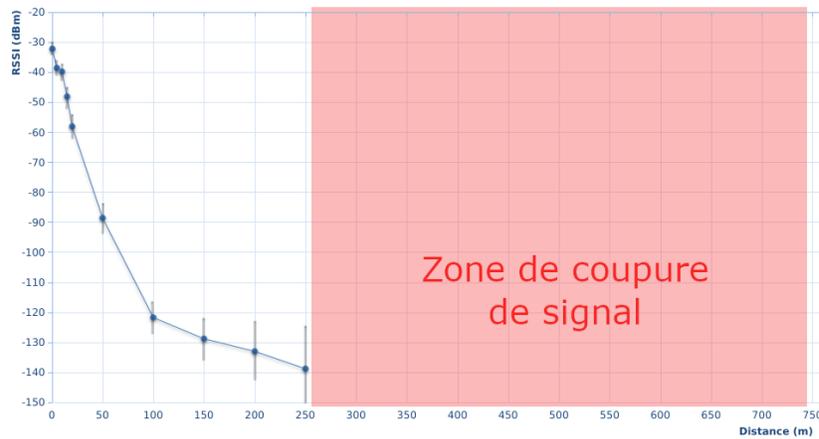


Fig. II.15 : RSSI en fonction de la distance, la technologie LoRa

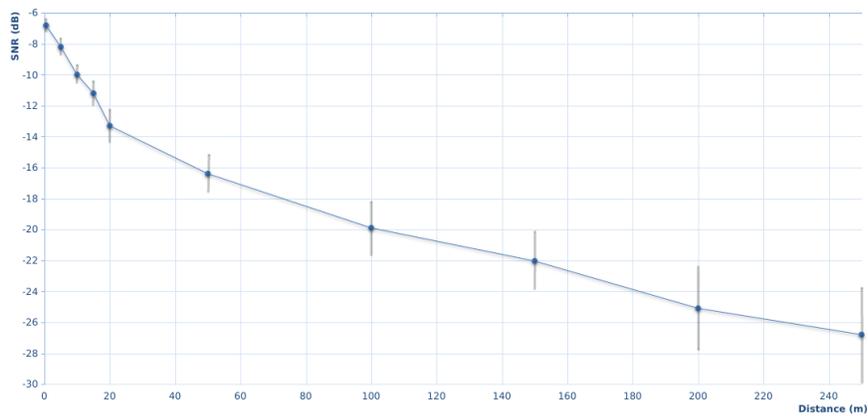


Fig. II.16 : SNR en fonction de la distance, la technologie LoRa

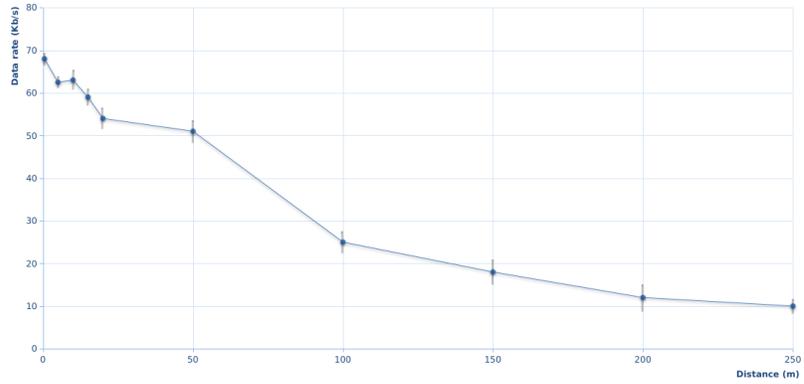


Fig. II.17 : Débit en fonction de la distance, la technologie LoRa

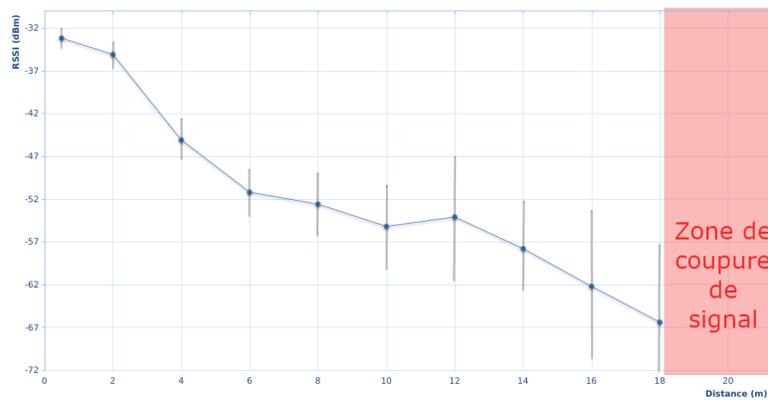


Fig. II.18 : RSSI en fonction de la distance, la technologie WiFi

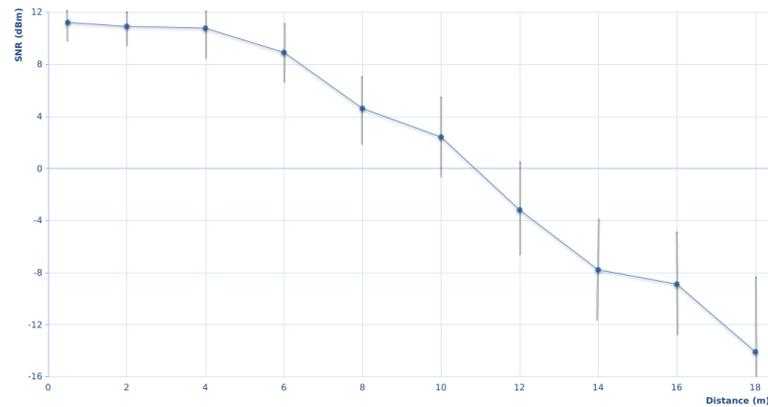


Fig. II.19 : SNR en fonction de la distance, la technologie Wifi

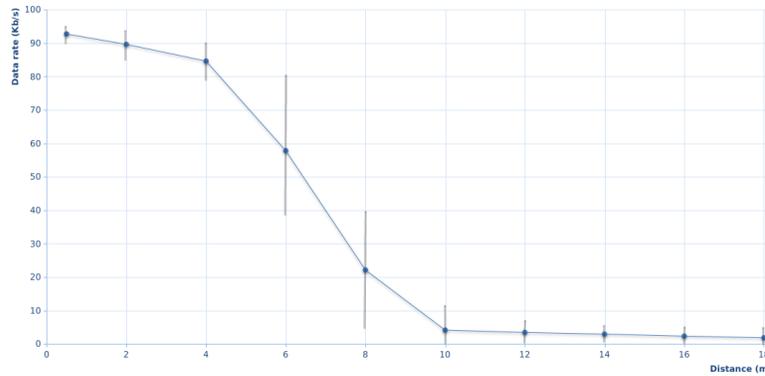


Fig. II.20 : Débit en fonction de la distance, la technologie Wifi

En ce qui concerne le RSSI, on remarque que le Wifi, Bluetooth et le Zigbee ont une affaiblissement de signal plus élevée par rapport au LoRa qui est un peu lent en terme d'affaiblissement et ce qui est logique car toute fois la fréquence augmente vers la FR-2 (Frequency Range 2) et FR-3 (Frequency Range 3) on trouve que le signal soit affaibli dans des distances plus courtes, par contre si on parle des fréquences basses de régions FR-1 (Frequency Range 1) que peuvent atteindre de distances plus éloignées.

Les écarts types significatifs pour le Bluetooth et le Wifi reflètent leur sensibilité accrue au bruit ambiant, entraînant des perturbations fréquentes des paquets de données. Le Bluetooth, en particulier, se distingue par sa complexité de captation et de calcul des paramètres de signal en raison des mécanismes de sécurisation. En revanche, le Wifi, plus accessible, présente également une variabilité marquée, soulignant sa vulnérabilité aux interférences.

Les résultats du Signal-to-Noise Ratio (SNR) démontrent une diminution générale avec l'augmentation de la distance, conformément aux attentes. Cependant, le Wifi et le Bluetooth affichent des SNR élevés à de courtes distances, soulignant leur capacité à émettre des signaux robustes face au bruit. Cette caractéristique, bien que bénéfique dans des scénarios de proximité, implique une consommation énergétique plus élevée par rapport au LoRa et au Zigbee, en particulier LoRa.

On a remarqué une chute de qualité de signal de Wifi et de Bluetooth le moment où il y a un obstacle d'eau ce qui rend l'impact de l'eau sur la qualité du signal, particulièrement observé dans le Wifi et le Bluetooth, met en lumière la sensibilité de ces technologies aux obstacles physiques, notamment aquatiques. À l'inverse, le Zigbee et le LoRa résistent davantage à ces interférences, soulignant leur adaptabilité à des environnements hostiles.

Si on focalise maintenant sur le débit, la résilience du LoRa face au bruit est clairement démontrée dans l'analyse du débit en fonction du SNR et la quantité de trafic reste stable même si l'énergie du bruit plus que l'énergie du signal ($SNR < 0$), cette propriété, résultant de sa modulation robuste, positionne le LoRa comme une option privilégiée pour des informations ou missions sensibles, malgré une légère surcharge (débit faible). Alors que le Wifi offre le débit le plus élevé, son efficacité diminue considérablement lorsque le SNR devient faible. Le Bluetooth, en raison de sa faible résistance au bruit, voit son débit décliner rapidement ce qui met le Bluetooth un mauvais choix pour des robots mobiles dans des lieux hostiles malgré qu'il fournisse une sécurisation élevée de l'information.

En conclusion, la technologie LoRa se positionne comme la solution privilégiée, offrant une immunité accrue au bruit, une stabilité notable et une consommation énergétique réduite. Sa capacité à maintenir des débits stables même dans des environnements bruyants en fait un choix optimal pour des applications où la fiabilité des transmissions est primordiale, et la sécurité fournie par LoRaWAN dans le protocole TCP/IP, renforcent sa position dans le choix de la technologie pour des robots mobiles collaboratifs. Pour des scénarios complexes, une solution hybride, combinant LoRa avec d'autres technologies telles que le Wifi ou le Zigbee en fonction des besoins, pourrait offrir une flexibilité supplémentaire. Ces résultats apportent des éclairages précieux pour guider le choix des technologies de transmission dans des contextes réels de robots mobiles collaboratifs évoluant dans des environnements hostiles.

II.7. Conclusion

En conclusion, cette analyse comparative des technologies de transmission Wi-Fi, Bluetooth, ZigBee et LoRa pour les robots mobiles en environnement hostile a mis en lumière leurs atouts et leurs limites. La recherche a mis en évidence les facteurs critiques que sont la force du signal, la résistance au bruit et l'efficacité du transfert de données, fournissant ainsi des informations précieuses pour les applications dans des environnements difficiles tels que les zones sinistrées, les installations industrielles et les terrains extérieurs.

Les résultats soulignent la robustesse de LoRa dans des conditions hostiles, démontrant ses performances supérieures en matière de maintien de la stabilité des communications, même face à des obstacles et des interférences. La faible consommation d'énergie et la portée étendue de la technologie la rendent particulièrement adaptée aux scénarios nécessitant une communication longue distance avec une consommation d'énergie minimale.

Pour l'avenir, les perspectives de cette étude suggèrent un avenir prometteur pour l'intégration de LoRa dans des applications robotiques mobiles au sein de sites hostiles. Sa résilience à la dégradation du signal et son adaptabilité à divers défis environnementaux positionnent LoRa comme un choix privilégié pour les missions robotiques collaboratives. De plus, l'exploration de solutions hybrides, combinant LoRa avec d'autres technologies basées sur des demandes contextuelles, ouvre la voie à des stratégies de communication adaptées et flexibles.

À mesure que la technologie évolue, des recherches plus approfondies peuvent se pencher sur les optimisations et les progrès de ces technologies de communication, en tenant compte des normes et protocoles émergents. De plus, des essais pratiques de mise en œuvre dans des scénarios réels seront cruciaux pour valider les résultats théoriques et garantir l'intégration transparente de ces technologies dans diverses applications robotiques. Cet article jette les bases des efforts futurs, guidant le développement de systèmes de communication permettant aux robots mobiles de naviguer et de collaborer efficacement face à l'adversité.

Chapitre III

Sécurité des réseaux IoT pour multirobots

Chapitre III : Sécurité des réseaux IoT pour multirobots

III.1. Introduction

Une formation cluster/essaim de microrobots est sujette d'interactions communicatives et de transfert de grands volumes de données. La topologie, dans ce sens n'est affectée que par le type du lien de connexion et des protocoles de gestion des données à échanger.

Certes, le fait que les machines connectées sur un segment réseau sont sur place ou déployés sur un site donné (et même déportés sur des sites non conjoints), n'influence en rien le fonction réseau mais représente une grande imposition sur le déploiement de sa stratégie de SÉCURITÉ. Vues les avancées enregistrées dans le domaine de l'utilisation de l'IA, les considérations se compliquent encore plus. Les machines traitant des données dans nos institutions sont les cibles privilégiées pour toute tentative d'intrusion. Dans notre thèse, nous avons validé cette implication sur deux cas de situations :

1. Réseau fait de machines fixes connectées en réseau IoT (cas d'équipement informatique réservé à la recherche scientifique) dans un laboratoire de recherche. À considérer que ce lieu, représente une vitale difficulté en relation avec les données échangées (Données relatives aux différents projets de recherche initiés avec des partenaires stratégiques).
2. Réseau monté à partir d'un cluster de machines mobiles (microrobots) sur un segment de lieux sensibles à une existence humaine (cas de microrobots, dédiés pour la recherche et l'investigation en milieu hostiles).

Les deux configurations, doivent être techniquement connectées à un réseau IoT et nécessitent le recours aux précautions de sécurité (Cybersécurité). Dans ce cas, la sécurité (Fig. III.1) est une exigence opérationnelle. Dans les sections qui suivent, la question est étudiée en détail.

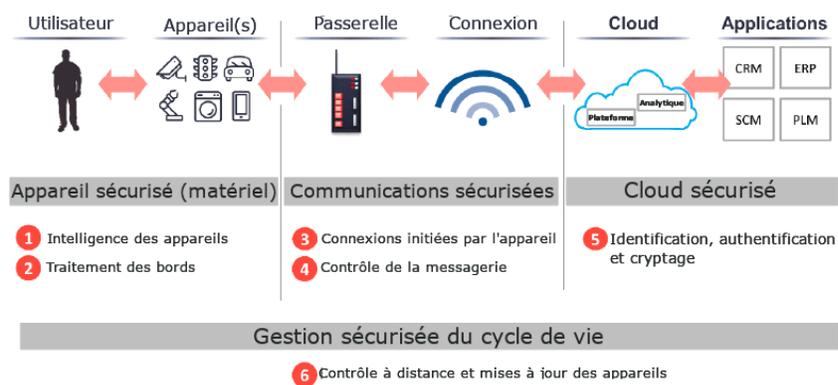


Fig. III.1 : Configuration de base d'un circuit IoT (bout-en-bout)

L'utilisation de l'Internet des objets (IoT) pour sécuriser les réseaux (Fig.III.2), notamment dans le contexte de robots mobiles connectés et collaboratifs, implique de tirer parti des appareils intelligents et interconnectés pour améliorer l'infrastructure de sécurité globale [173, 174].

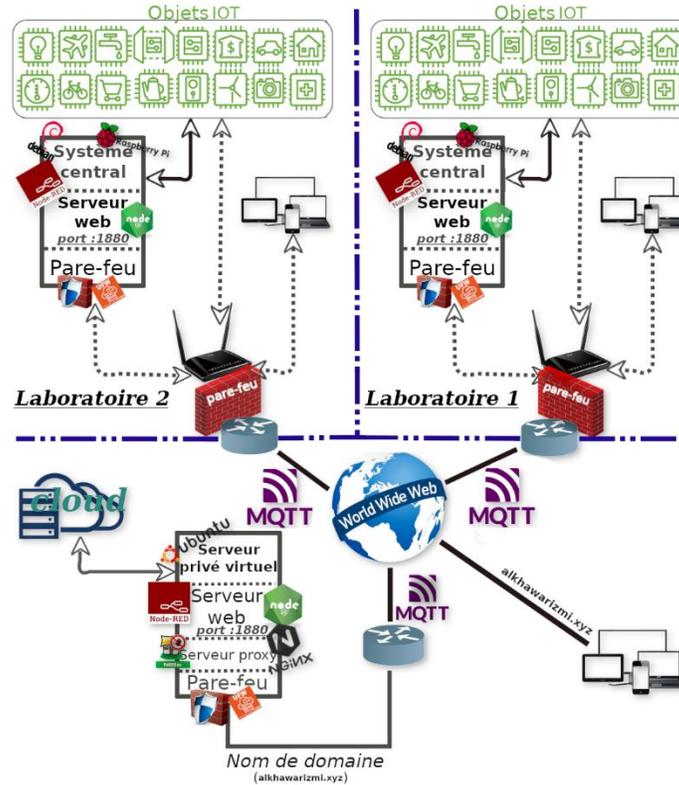


Fig. III.2 : Un système réseau IoT sécurisé pour la gestion des laboratoires [175]

À l'ère de l'Industrie 4.0, où l'automatisation et la connectivité sont primordiales, la sécurisation des réseaux est devenue une préoccupation majeure, en particulier dans les environnements où les robots mobiles collaboratifs jouent un rôle central [176, 177]. L'intégration de l'Internet des objets (IoT) dans les stratégies de sécurité des réseaux s'est avérée être une approche transformatrice, offrant des capacités avancées pour protéger les systèmes interconnectés.

III.2. IoT et la sécurisation des réseaux

Les appareils IoT servent de capteurs et d'actionneurs intelligents qui surveillent et collectent en permanence des données de l'environnement (Figures III.3 et III.4). Dans le contexte de la sécurité des réseaux, ces appareils agissent comme des gardiens distribués, évaluant en permanence l'état du réseau et de ses composants connectés. Ils fournissent des informations en temps réel sur les vulnérabilités potentielles, les activités anormales et les menaces émergentes [173, 178, 179].

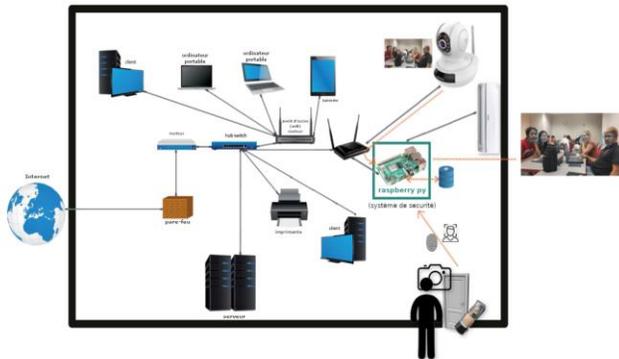


Fig. III.3 : Un schéma de déploiement représente un réseau IoT d'un laboratoire avec des objets [175]

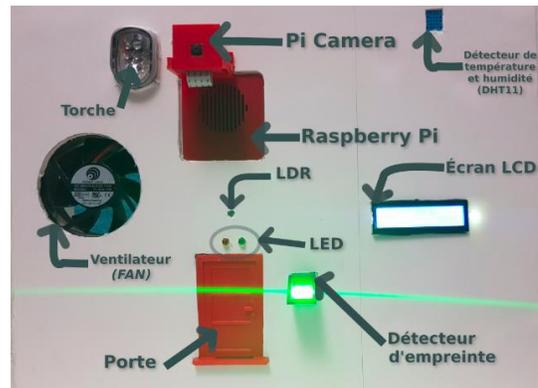


Fig. III.4 : Un prototype réalisé de processus de contrôle d'accès d'un réseau IoT avec quelques objets que le réseau renferme [175]

III.3. Sécuriser les robots mobiles collaboratifs avec l'IoT

Les robots mobiles collaboratifs, souvent déployés dans des environnements industriels et logistiques, font partie intégrante des opérations rationalisées. Ces robots collaborent entre eux et avec des humains, partageant des données et coordonnant les tâches pour une efficacité optimale. Cependant, leur intégration dans les réseaux expose à des problèmes de sécurité potentiels, ce qui rend crucial la mise en œuvre de mesures robustes pour protéger ces systèmes interconnectés [180-183].

III.4. Contrôle continu

Les appareils IoT intégrés dans des robots mobiles collaboratifs (Fig.I.17) surveillent en permanence leur environnement et leurs systèmes internes. Cela comprend l'évaluation de l'intégrité des canaux de communication, la détection des tentatives d'accès non autorisées et l'identification de tout modèle inhabituel dans les transmissions de données [184-186].

- Pour faire l'intégration des capteurs, les robots mobiles collaboratifs sont équipés de divers capteurs, tels que des caméras, des lidars et des capteurs de proximité. Ces capteurs sont compatibles IoT, ce qui leur permet de transmettre des données en temps réel à un système de surveillance central.
- Les données des capteurs sont transmises à l'aide de protocoles sécurisés et efficaces tels que MQTT (Message Queuing Telemetry Transport) ou CoAP (Constrained Application Protocol), garantissant une faible latence et une communication fiable.
- Les appareils IoT sur les robots utilisent des algorithmes d'apprentissage automatique pour la détection des anomalies. Ces algorithmes analysent les données des capteurs pour établir un comportement de base et identifier les écarts pouvant indiquer des menaces potentielles pour la sécurité ou des conditions de fonctionnement anormales.

III.5. Détection et réponse aux menaces

Les systèmes de sécurité compatibles IoT analysent les données collectées par les robots mobiles collaboratifs pour détecter les menaces de sécurité potentielles. Les algorithmes d'apprentissage automatique peuvent identifier des modèles révélateurs de cyberattaques ou d'accès non autorisés, déclenchant ainsi des mécanismes de réponse immédiats pour atténuer les risques [187-189].

- Si on parle de l'analyse comportementale, les modèles d'apprentissage automatique analysent en permanence le comportement des robots mobiles collaboratifs. Tout écart par rapport au comportement attendu, tel que des changements soudains dans les schémas de mouvement ou des lectures inattendues des capteurs, déclenche une alerte pour une enquête plus approfondie.
- Pour la gestion des informations et des événements de sécurité (SIEM), le système SIEM regroupe et corrèle les données de divers appareils IoT sur les robots. Il analyse les journaux et les événements en temps réel pour identifier les incidents de sécurité potentiels, offrant ainsi une vue centralisée de l'état de sécurité du réseau.
- En cas de menace détectée, des mécanismes de réponse automatisés peuvent être déclenchés. Cela peut inclure l'isolement du robot concerné du réseau, la notification aux

administrateurs ou l'ajustement dynamique des politiques de sécurité pour atténuer la menace.

III.6. Cryptage des données

Pour sécuriser la communication entre les robots mobiles collaboratifs et le réseau, les appareils IoT mettent en œuvre des protocoles de cryptage forts. Cela garantit que les données transmises entre les appareils et les serveurs restent confidentielles et ne sont pas susceptibles d'être interceptées ou falsifiées [190, 191].

- **Chiffrement de bout en bout** : La communication entre les robots mobiles collaboratifs et le réseau est sécurisée par un cryptage de bout en bout. Les protocoles Transport Layer Security (TLS) ou Datagram Transport Layer Security (DTLS) sont couramment utilisés pour garantir la confidentialité et l'intégrité des données.
- **Gestion des clés** : Des pratiques robustes de gestion des clés sont essentielles pour maintenir la sécurité du chiffrement. Un stockage et une distribution sécurisés des clés de chiffrement, ainsi qu'une rotation régulière des clés, sont mis en œuvre pour contrecarrer les accès non autorisés.

III.7. Gestion à distance et mises à jour

L'IoT facilite la gestion à distance des protocoles de sécurité et des mises à jour logicielles. Ceci est crucial pour maintenir la résilience des défenses du réseau contre l'évolution des menaces. Des mises à jour régulières peuvent être déployées sur les robots mobiles collaboratifs pour corriger les vulnérabilités et améliorer la sécurité globale [192-194].

- **Mises à jour en direct (OTA)** : Les robots mobiles collaboratifs compatibles IoT prennent en charge les mises à jour OTA, permettant ainsi le déploiement à distance des correctifs micrologiciels et logiciels. Cela garantit que les vulnérabilités de sécurité sont rapidement corrigées sans nécessiter d'intervention physique.
- **Plateformes de gestion des appareils** : Les plates-formes de gestion centralisée des appareils permettent aux administrateurs de surveiller et de gérer à distance les configurations de sécurité des robots mobiles collaboratifs. Cela comprend l'application des politiques de sécurité, la mise à jour des logiciels de sécurité et la réponse aux incidents de sécurité.

- Démarrage sécurisé et intégrité du micrologiciel : Des mécanismes de démarrage sécurisé sont mis en œuvre pour garantir que seul le micrologiciel authentifié et autorisé peut être chargé sur les robots. Cela évite les modifications malveillantes du micrologiciel qui pourraient compromettre la sécurité des appareils.

III.8. Authentification et contrôle d'accès

L'utilisation de l'IoT pour la sécurité du réseau implique la mise en œuvre de mécanismes robustes d'authentification et de contrôle d'accès. Les robots mobiles collaboratifs bénéficient de privilèges d'accès basés sur des informations d'identification vérifiées, et tout écart par rapport à la norme déclenche des protocoles de sécurité pour empêcher les actions non autorisées [186, 195-199].

- Infrastructure à clé publique (PKI), la mise en œuvre d'une PKI garantit une authentification sécurisée entre les robots mobiles collaboratifs et le réseau. Chaque appareil se voit attribuer un certificat numérique unique et des protocoles d'authentification mutuelle sont utilisés pour vérifier l'identité des robots et des composants du réseau.
- Contrôle d'accès basé sur les rôles (RBAC), les mécanismes RBAC sont utilisés pour définir et gérer les privilèges d'accès. Différents rôles sont attribués aux robots mobiles collaboratifs en fonction de leurs fonctions, et les autorisations d'accès sont strictement appliquées pour empêcher les actions non autorisées.
- Authentification multi-facteur (MFA), améliorant la sécurité de l'authentification, MFA est mis en œuvre pour exiger plusieurs formes de vérification avant d'accorder l'accès. Cela pourrait inclure une combinaison de mots de passe, de données biométriques et d'authentification basée sur l'appareil.

Listes de contrôle d'accès (ACL) sont très importantes pour le contrôle des paquets dans les réseaux IoT, les listes de contrôle d'accès (ACL) jouent un rôle crucial dans le contrôle du flux de paquets au sein des réseaux IoT, en particulier pour les robots mobiles collaboratifs. Les ACL définissent des règles qui déterminent quels périphériques ou entités sont autorisés ou refusés à accéder à des ressources réseau spécifiques. Dans le contexte des robots mobiles collaboratifs et de l'IoT, les ACL sont utilisées pour un contrôle précis des interactions des paquets. Les ACL peuvent être configurées pour identifier et authentifier les appareils en fonction de leurs identifiants uniques, tels que les adresses MAC ou les certificats numériques. Cela garantit que seuls les

appareils autorisés, y compris les robots mobiles, sont autorisés à communiquer au sein du réseau. Les ACL définissent des règles spécifiant quels types de paquets sont autorisés ou refusés en fonction de critères tels que les adresses source et de destination, les ports et les protocoles. Pour les robots mobiles, cela permet de contrôler le flux de données entre les robots et les autres composants du réseau, empêchant ainsi les communications non autorisées et ils peuvent être intégrées aux mécanismes RBAC, améliorant ainsi le cadre de contrôle d'accès. Cela permet aux administrateurs de définir des autorisations d'accès en fonction des rôles spécifiques attribués aux robots mobiles collaboratifs, garantissant ainsi que chaque robot a accès uniquement aux ressources requises pour ses tâches désignées.

Les pare-feux agissent comme une ligne de défense essentielle pour protéger les réseaux IoT, y compris ceux prenant en charge les robots mobiles collaboratifs, contre tout accès externe non autorisé. Les pare-feux établissent une barrière entre le réseau interne et les entités externes, contrôlant le trafic entrant et sortant sur la base de règles de sécurité prédéfinies.

- Filtrage des entrées et des sorties : Les pare-feu mettent en œuvre un filtrage des entrées pour contrôler le trafic entrant, garantissant que seules les communications légitimes et autorisées sont autorisées sur le réseau. Le filtrage de sortie empêche les données sensibles de quitter le réseau sans autorisation appropriée.
- Inspection avec état : Les pare-feu modernes utilisent souvent des techniques d'inspection dynamique, qui suivent l'état des connexions actives. Cela permet d'empêcher les tentatives d'accès non autorisées en garantissant que les paquets entrants font partie de connexions établies et légitimes.
- Traduction d'adresses réseau (NAT) : NAT est souvent utilisé conjointement avec des pare-feux pour masquer les adresses IP internes des entités externes. Cela ajoute une couche de sécurité supplémentaire en obscurcissant la structure du réseau interne et en empêchant les attaquants potentiels de cibler directement des appareils individuels.
- Prise en charge du réseau privé virtuel (VPN) : Pour une communication sécurisée entre les robots mobiles et le réseau central, les pare-feux peuvent prendre en charge les VPN. Les VPN chiffrent le canal de communication, garantissant que les données transmises entre les robots et le réseau restent confidentielles et sécurisées, même lorsqu'elles traversent des réseaux externes.

L'intégration d'ACL pour le contrôle des paquets dans les réseaux IoT et le déploiement de pare-feu pour protéger contre les accès externes font partie intégrante du cadre d'authentification et de contrôle d'accès. Ces mesures garantissent que seuls les appareils autorisés, tels que les robots mobiles collaboratifs, peuvent communiquer au sein du réseau, préservant ainsi l'intégrité et la sécurité de l'écosystème IoT. En combinant ACL et pare-feu, les organisations peuvent établir un mécanisme de défense robuste contre les accès non autorisés et les cybermenaces potentielles.

III.9. Conclusion

La sécurité des réseaux IoT est essentielle au bon fonctionnement des systèmes connectés. Les appareils IoT collectent et partagent des données sensibles, ce qui les rend vulnérables aux cyberattaques. Pour renforcer la sécurité, il est essentiel d'utiliser des protocoles de communication cryptés et des mécanismes d'authentification robustes. La mise en œuvre de pare-feu et de systèmes de détection d'intrusion permet également d'empêcher les accès non autorisés. Les fabricants devraient intégrer des mises à jour logicielles automatiques pour corriger les vulnérabilités émergentes. De plus, éduquer les utilisateurs finaux sur les bonnes pratiques de sécurité et encourager l'utilisation de mots de passe forts renforce la protection globale. Enfin, la collaboration entre les acteurs du secteur pour partager des informations sur les menaces permet d'anticiper et de contrer les attaques potentielles, garantissant ainsi la sécurité des réseaux IoT.

Chapitre IV

Allocation de tâches pour multirobot (MRTA)

Chapitre IV : Allocation de tâches pour multirobot (MRTA)

IV.1. Introduction

Le problème concerne l'allocation en temps réel de tâches imprévisibles et inconnues arrivant à des heures et des lieux inconnus. L'occurrence de tâche est dynamique et imprévisible avec différents niveaux d'importance de chaque tâche et doit être détectée aussi rapidement que possible. Des exemples de telles tâches comprennent la surveillance (collecte d'informations sur les objets désirés, [200, 201]), la surveillance de la sécurité (prévention du vol de marchandises et des menaces, [202]), la surveillance des incendies (allocation et protection contre les incendies de forêt, [203, 204]), entre autres. Les capteurs doivent être affectés aux tâches le plus rapidement possible. Le but est d'attribuer à chaque capteur une tâche appropriée à un moment approprié (Fig.IV.1) et de s'assurer que toutes les tâches sont accomplies en un minimum de temps. Le système comprend plusieurs capteurs capables d'exécuter chaque tâche avec des performances différentes. Les performances sensorielles sont définies a priori en fonction des caractéristiques du capteur, à savoir la distance d'allocation, la résolution et le temps de réponse.

Chaque capteur ne peut être affecté qu'à une seule tâche à un moment donné et peut être réaffecté à une autre tâche à tout moment. La priorité d'une tâche est une valeur scalaire spécifique à l'application, où une valeur de priorité plus élevée représente une tâche qui a une importance plus élevée et doit être traitée plus rapidement que les autres tâches. Les tâches de priorité plus élevée ont également un avantage plus élevé pour les accomplir.

Il n'y a pas de limitation du nombre de capteurs pouvant être alloués à une seule tâche [205] comme il n'y a pas d'exigence d'affecter aux nœuds (robots/cobots comme exemple) le même ensemble de capteurs. Cela élimine le conflit d'exiger la même tâche par plusieurs capteurs.

Une fois qu'un capteur trouve une tâche, il informe les capteurs (nœuds) voisins de la tâche trouvée et de ses paramètres, en utilisant une communication par diffusion (comme dans [206-209]). Ce message est ensuite transmis par ces capteurs sur l'ensemble du réseau, comme dans Ducatelle et al. [206]. A partir de ce moment, les capteurs sont conscients de la tâche détectée. Même si les capteurs utilisent la communication par diffusion pour partager l'emplacement estimé et la priorité des tâches, l'attribution des tâches est effectuée de manière décentralisée. Chaque nœud prend une décision autonome basée sur les informations qu'il a reçues [210].

Lorsque plusieurs tâches nécessitent les mêmes capteurs, l'attribution dépend de la disponibilité et des performances des capteurs, de la distance physique des capteurs par rapport aux tâches et des priorités des tâches individuelles.

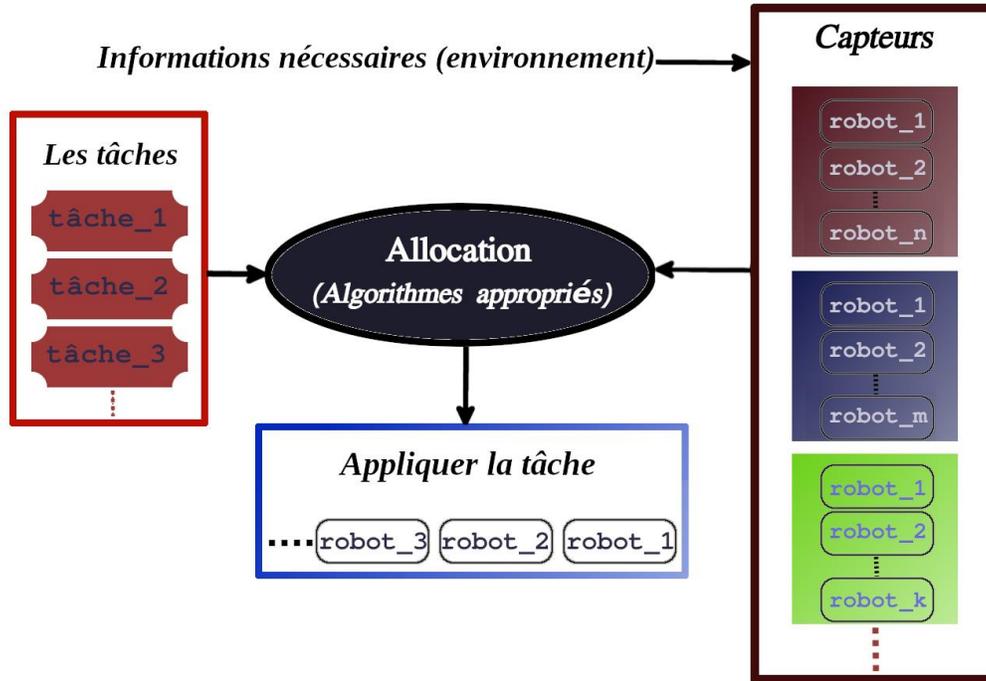


Fig. IV.1 : Schéma d'allocation des tâches d'un système multi-capteur

IV.2. Allocation de tâche selon l'architecture du réseau multi-nœud

IV.2.1. Allocation centralisée des tâches multi-nœuds d'agents

Les approches centralisées reposent généralement sur un nœud agent central qui coordonne l'attribution des autres nœuds [211]. Ces approches sont les mieux adaptées aux applications où les équipes sont petites et où les informations globales sur les tâches sont facilement disponibles [212].

Le principal avantage des systèmes centralisés est que dans certains problèmes, les solutions obtenues peuvent être optimales ou très proches de l'optimal [211, 213]. Cependant, cela suppose que les informations des différents nœuds agent sont suffisamment précises.

Mais, les systèmes centralisés sont moins robustes puisqu'ils reposent sur un élément central qui calcule l'allocation optimale et en cas de défaillance de cet élément, l'ensemble du système échoue [214, 215]. D'autres problèmes liés à cette approche incluent les limitations de la couverture de communication (c.-à-d. La diffusion de messages sur de longues distances à partir

d'un nœud agent centralisé, où de grandes équipes d'agents couvrent un grand espace) et l'évolutivité (c.-à-d., Nécessite de modifier les algorithmes pour l'ajout/la soustraction de capteurs pendant le fonctionnement, [211, 216]).

Pour le cas d'un problème de la conception d'un système robuste et évolutif avec un grand nombre d'agents et de tâches. On considère que ces agents opèrent dans des environnements inconnus sans savoir a priori quand, où et quelle tâche entrera ou sortira de l'environnement. Ce problème ne peut pas être traité par une approche centralisée [205].

IV.2.2. Allocation de tâches multi-nœuds d'agents décentralisée

Dans les approches décentralisées autoorganisées, chaque nœud agent prend ses propres décisions sans considération majeure des autres agents [217]. Ils comprennent des méthodes basées sur l'intelligence des essaims, comme la colonie de fourmis ou les abeilles distribuées, qui permettent d'obtenir un comportement collectif complexe à partir des interactions locales de nombreux individus avec un comportement simple. Dans ces approches, les capteurs utilisent les connaissances locales et partagent des informations entre eux [216, 218]. Dans de tels systèmes, les capteurs collaborent afin d'atteindre un objectif global.

Des approches bio-inspirées de l'intelligence des essaims, influencées par le comportement émergent des insectes sociaux, tels que les fourmis, les abeilles et les termites, ainsi que d'autres sociétés animales, telles que les volées d'oiseaux ou les bancs de poissons, ont été utilisées pour modéliser le comportement des systèmes multi-nœuds agent intelligents [216, 219-221]. Dans les algorithmes d'intelligence d'essaim [222-224], les nœuds agent coopérants interagissent en échangeant des informations disponibles localement, de sorte que l'objectif global est obtenu plus efficacement qu'il ne le serait par des agents qui exécutent des tâches individuellement [221]. Ces algorithmes ont été appliqués à des applications d'exploration multi-agents et de formation de chemins [225], d'optimisation énergétique dans des réseaux de capteurs [226, 227], de déploiement multisites [228], d'optimisation de calcul parallèle [229], d'attribution de tâches [230], transport coopératif et acheminement de véhicules [231, 232], sélection de caractéristiques [233], allocation d'intrus [234], allocation de ressources [235], allocation de tâches multi-robot et applications de suivi [230, 236], problème de sac à dos (knapsack) [237], ou transport coopératif et récupération des proies [231]. Les inconvénients du contrôle intelligent des essaims comprennent des conflits entre les membres des essaims, des activités redondantes et une réponse globale lente à un changement de l'environnement [223, 238] en général, ils ne fournissent que des

solutions sous-optimales [213]. L'algorithme DBA (Distributed Bees Algorithm) ; comme exemple ; convient à la mise en œuvre dans un système multi-nœuds d'agents et est également évolutif avec une faible consommation de calcul [208]. Cette topologie décentralisée inhérente aux systèmes multi-nœuds d'agents bio-inspirés leur donne la capacité d'être autonomes, évolutifs, robustes et adaptatifs aux changements de leur environnement et aux contraintes du monde réel [213]. Ces approches, bien qu'adaptées à l'application multi-nœud d'agents, n'ont pas été mises en œuvre pour allouer des agents statiques hétérogènes à des tâches dynamiques [205].

IV.2.3. Allocation de tâches multi-nœuds d'agents hybride

Les approches hybrides utilisent des agents décentralisés pour contrôler et exploiter les points de centralisation sous forme d'enchères pour produire des allocations [239, 240]. Ils incluent des méthodes intentionnelles telles que des algorithmes basés sur le problème du marché (Market-based) et des protocoles de réseau contractuel. Dans de tels systèmes, les agents intéressés participent à une économie de marché virtuelle et répartissent les tâches au moyen de procédures d'appel d'offres.

IV.3. Travaux connexes pour des systèmes d'allocation de tâches multi-nœuds d'agents

Tkach et Edan [205] caractérise les systèmes fondés sur l'allocation de tâches entre un ensemble de nœuds appelés agents en trois techniques qui renferment des algorithmes collaboratifs pour la distribution de tâches afin de maximiser le nombre de tâches accomplies avec efficacité, moins d'erreurs et selon le type de tâches et du système réseau d'agents/capteurs.

IV.3.1. Méta heuristique (Meta Heuristics)

Les méta-heuristiques ont généralement un comportement itératif et sont apparues ces dernières années comme des alternatives réussies aux approches classiques pour résoudre des problèmes d'optimisation complexes dans un temps raisonnable [241-243]. Les algorithmes méta heuristiques incluent l'intelligence d'essaim, le calcul évolutif, le recuit simulé, la recherche tabou et, l'escalade (Hill climbing) entre autres [244, 245].

Le calcul évolutif comprend des algorithmes d'optimisation inspirés de l'évolution darwinienne. Dans ces algorithmes, un ensemble de solutions est appelé une population. Chaque itération est utilisée pour créer une nouvelle génération à partir de la population actuelle. Cette approche comporte généralement deux opérateurs, appelés recombinaison ou croisement, qui recombine deux ou plusieurs individus et des mutations, ce qui modifie un seul individu [246]. Sur la base de la fonction objective, les individus de la génération actuelle sont sélectionnés pour la

recombinaison et la mutation, ou pour être répétés pour la génération suivante sans aucun changement, tandis que les individus avec une meilleure forme physique ont une probabilité plus élevée d'être sélectionnés. Les algorithmes de calcul évolutifs comprennent la programmation évolutive [247], les stratégies évolutives [248] et les algorithmes génétiques [249].

La recherche tabou, elle introduit par Glover [250], utilise l'historique de recherche pour explorer l'espace des solutions. Cet algorithme applique la meilleure recherche locale d'amélioration avec mémoire pour éviter les cas de l'optimum local. La mémoire est implémentée sous la forme d'une liste taboue qui garde la trace des solutions les plus récemment visitées et interdit les déplacements vers elles. À chaque itération, la meilleure solution est choisie dans l'ensemble autorisé comme nouvelle solution actuelle.

La méthode d'escalade est un algorithme d'optimisation de la recherche locale [251-253]. Il s'agit d'un algorithme itératif qui tente de trouver une meilleure solution en apportant des modifications à la solution actuelle. Si le changement produit une meilleure solution, il est sélectionné pour être modifié davantage jusqu'à ce qu'une meilleure solution soit obtenue [253]. La méthode d'escalade peut souvent produire un meilleur résultat lorsque le temps disponible pour effectuer une recherche est limité [205]. C'est bon pour trouver des cas de l'optimum local, mais il n'est pas nécessairement garanti de trouver la solution optimale.

Le recuit simulé ou Simulated Annealing introduit dans le domaine de l'optimisation combinatoire par Kirk-Patrick et al. [254], est une méta-heuristique de pointe pour approximer l'optimisation globale dans un grand domaine [253] et l'une des premières ayant une stratégie pour échapper aux cas de l'optimum local. Il simule un processus de recuit physique en métallurgie, où les particules d'un solide s'organisent dans un équilibre thermique, une technique impliquant le chauffage et le refroidissement contrôlé d'un matériau pour augmenter la taille de ses cristaux et réduire leurs défauts. Le recuit simulé interprète un refroidissement lent comme une faible diminution de la probabilité d'accepter de pires solutions lors de l'exploration de l'espace des solutions. Accepter de pires solutions est une propriété fondamentale de la méta-heuristique car elle permet une recherche plus approfondie de la solution optimale (explore l'espace des solutions), réduisant les chances d'être convergé vers des cas de l'optimum local. Les méthodes heuristiques combinent intelligemment des concepts différents pour explorer l'espace de recherche afin de trouver des solutions quasi optimales [255] et de permettre de s'échapper des cas de l'optimum

local. Cependant, ils ne garantissent pas toujours de trouver une solution globalement optimale [244].

IV.3.2. Intelligence d'une configuration en essaim

Les systèmes basés sur des essaims sont généralement constitués d'une population d'agents simples (entités système qui prennent des décisions sur leur environnement local) interagissant localement les uns avec les autres et avec leur environnement pour atteindre un objectif commun [221]. Ces approches simulent les comportements d'insectes et d'animaux pour assigner des tâches aux agents. Les méthodes d'intelligence en essaim [256] sont principalement utilisées pour les systèmes d'agents dans des environnements inconnus et l'optimisation pour l'attribution de tâches, le regroupement, la navigation et la formation de chemins [219, 257]. Puisque la collaboration de groupe entre les individus est distribuée, un défaut individuel n'affecte pas la solution globale [217, 258]. Intelligence computationnelle en essaim (Computational Swarm Intelligence) représente le groupe d'algorithmes d'intelligence computationnelle qui modélisent les comportements d'essaimage dans la nature. L'intelligence computationnelle appartient au domaine plus large de l'intelligence artificielle et des paradigmes qui se rapportent à une sorte de système biologique ou naturel [259]. Les méthodes d'intelligence en essaim ont une robustesse et une évolutivité élevées et conviennent aux systèmes multi-agents distribués [260].

Les avantages de la collaboration peuvent être importants dans les situations où la connaissance globale de l'environnement n'existe pas. Les agents au sein du groupe interagissent en échangeant des informations disponibles localement de sorte que l'objectif global puisse être obtenu plus efficacement qu'il ne le serait par un seul agent. Le groupe d'agents agissant de cette manière peut être appelé un essaim. Le comportement de résolution de problèmes qui émerge des interactions de tels agents est appelé l'intelligence en essaim (Swarm Intelligence). Les modèles algorithmiques d'un tel comportement sont appelés l'intelligence computationnelle en essaim (Computational Swarm Intelligence) [261].

Les exemples de comportement collectif dans la nature sont nombreux. Ils reposent sur un échange direct ou indirect d'informations sur l'environnement entre les membres de l'essaim [74]. Si les règles régissant les interactions au niveau local sont généralement faciles à décrire, le résultat d'un tel comportement est difficile à prévoir [262]. Cependant, grâce à la collaboration, les essaims dans la nature sont capables de résoudre des problèmes complexes qui sont cruciaux pour la survie dans un environnement en évolution dynamique [223, 263]. Il existe de nombreux exemples de

comportement d'essaimage dans la littérature, car il a toujours intrigué les scientifiques pour sa meilleure compréhension. Les fourmis communiquent en déposant des phéromones sur leur route afin de trouver le chemin le plus court vers une source de nourriture en utilisant un comportement de recherche de nourriture [219, 256, 264].

Les termites coopèrent régulièrement pour construire des structures de nids complexes sans avoir aucune connaissance globale de l'environnement [221, 264]. Les abeilles utilisent la danse pour recruter d'autres membres de l'essaim afin de les suivre jusqu'à l'endroit du champ riche en nectar [265]. Les oiseaux se rassemblent en troupes et les poissons forment des bancs pour avoir de meilleures chances de survie contre les prédateurs [266]. Les bactéries échangent localement des informations sur leur environnement, etc. Pham et coll. [267] ont développé un algorithme des abeilles (BA pour Bees algorithm), qui, dans sa version de base ; effectue une recherche aléatoire combinée à une recherche de voisinage et peut être utilisé pour l'optimisation. L'algorithme exploite le concept de la piste de danse centrale afin de sélectionner les sites les plus adaptés, mais aucune communication directe entre les membres de l'essaim n'existe. Le recrutement des abeilles peut se faire de manière déterministe en fonction des valeurs de fitness associées aux sites, ou ces valeurs de fitness peuvent être utilisées pour déterminer la probabilité que les abeilles soient sélectionnées. Avec le scoutisme, ce recrutement différentiel est une opération clé de la BA. L'algorithme est exécuté par itérations, et il est arrêté lorsque la solution est trouvée dans la marge d'erreur fournie ou lorsque le nombre maximal d'itérations est atteint. Bailis et Coll. [268] ont proposé un modèle de recherche de nourriture pour la colonie d'abeilles afin d'étudier la valeur du partage d'informations sur la position des sources de nourriture dans différents environnements. Karaboga et Akay [269] ont proposé un algorithme d'optimisation basé sur les colonies d'abeilles appelé *Artificial Bee Colony* (ABC). L'algorithme utilise le recrutement basé sur les valeurs de fitness des sources de nourriture et applique la recherche de quartier pour des améliorations de solutions. Jevtić [221] a proposé un algorithme d'abeilles distribuées pour l'attribution de tâches dans un essaim de robots. Cet algorithme assigne les robots d'un essaim aux tâches trouvées de telle sorte que la distribution finale soit proportionnelle aux qualités des tâches. Les robots ont été conçus pour utiliser la communication par diffusion pour informer les autres robots de la gamme de l'emplacement estimé et de la qualité de la tâche trouvée. Peu d'autres algorithmes inspirés du comportement des abeilles sont apparus dans la littérature, tels que BeeHive [270], BCO [271],

Virtual Bee Algorithm [272], HBMO [273], etc., et ils ont surtout été appliqués à la résolution de problèmes d'optimisation combinatoire.

IV.3.3. Approches basées sur le marché (Market-Based)

Les approches basées sur le marché ont gagné en popularité ces dernières années, principalement parce qu'elles offrent un bon compromis entre trouver une bonne allocation et la vitesse de communication [217]. Les problèmes économiques ont impliqué les humains à travers l'histoire pour rechercher le profit individuel. Cette poursuite a conduit au formalisme des principes de l'économie de marché. Ces principes ont inspiré des approches informatiques basées sur le marché qui sont appliquées à l'allocation multi-nœuds d'agents [274]. La première méthode d'application des principes fondés sur le marché était le Contract Net Protocol ou CNP [275]. Il attribue des tâches aux ressources par la négociation de contrats. Il établit une économie virtuelle avec des agents en tant qu'acheteurs, des tâches en tant que produits et de l'argent virtuel en tant que monnaie. Des approches basées sur le marché ont été appliquées à l'allocation des ressources sur les ordinateurs [276-278], multi-agents [279], [280] et la coordination multirobot [281-284]. Dans les systèmes basés sur le marché, les agents agissent pour maximiser leur profit individuel et améliorer simultanément l'efficacité de l'équipe. Ils sont payés en fonction de leur offre pour les tâches qu'ils accomplissent et doivent payer la valeur des ressources qu'ils consomment. Les enchères sont gérées par le commissaire-priseur. Le commissaire-priseur peut être un agent de contrôle centralisé ou l'un des agents. Dans chaque vente aux enchères, les agents se font concurrence en soumissionnant pour être affectés à des tâches et maximiser leurs profits. La nouvelle allocation obtenue par appel d'offres peut aboutir à de meilleures solutions. L'un des premiers algorithmes distribués basés sur le marché était [285] proposé pour la coopération entre plusieurs robots [286]. Dans cet algorithme, les robots utilisent un principe en deux étapes pour calculer le coût d'une tâche en considérant la suivante afin d'augmenter l'efficacité de la solution. Dans MURDOCH [287, 288], les robots ne participent pas aux enchères pendant l'exécution des tâches, et les tâches ne sont allouées qu'aux robots inactifs. Il a été rapporté qu'il en résulte des solutions moins efficaces car il ne tient pas compte des tâches futures et utilise purement une approche gloutonne (Greedy Approach). Cependant, l'avantage de MURDOCH est qu'il utilise moins de communication. TraderBots [289] est un système basé sur le marché conçu pour fonctionner dans des environnements dynamiques. Ce système applique la planification pour allouer plusieurs tâches à chaque robot et permet d'enchérir sur plusieurs tâches simultanément. Il

a été rapporté qu'il faisait des allocations efficaces malgré des pannes partielles des robots et des communications (Dias et al. [290]). Ramchurn et al. [291] présentent une formation de coalition avec problème de contraintes spatiales et temporelles (CFSTP). Les agents forment des coalitions pour travailler ensemble sur des tâches avec des contraintes spatiales et des délais. Les agents ont des délais serrés et partagent les tâches à des taux non additifs pour maximiser le nombre de tâches terminées. CFSTP a été résolu à l'origine en utilisant formation d'une coalition avec anticipation (CFLA) [291] qui applique deux heuristiques, l'une, allouer la plus petite coalition possible, et l'autre, maximiser le nombre d'autres tâches qui peuvent être accomplies avant la date limite de la prochaine étape de temps. Un algorithme DCOP à somme maximale distribuée a été utilisé pour résoudre CFLA [292-294]. Walsh et Wellman [295] ont présenté un mécanisme pour atteindre un objectif similaire via un protocole de marché distribué pour allouer des tâches à des agents qui se disputent des ressources limitées. Les agents négocient des tâches à des prix déterminés par un protocole d'enchères. Dans Jones et al. [280], un mécanisme fondé sur le marché a été appliqué à un problème réel de lutte contre les incendies. Dans ces travaux, les incidents d'incendie définis comme des tâches ont été attribués aux unités de lutte contre l'incendie selon l'ordre dans lequel les incendies ont été découverts ; chaque incendie avait un commissaire-priseur, et les agents ont calculé le changement d'utilité pour l'attribution d'une tâche, en utilisant un ordre optimal ou presque optimal, et ont soumis ce changement comme une offre. Le commissaire-priseur a confié avec avidité et irrévocablement la tâche à l'agent ayant l'offre la plus élevée. Une fois qu'un incendie a été attribué à un agent, celui-ci en était responsable jusqu'à ce qu'il soit éteint.

Plusieurs travaux basés sur le marché se sont concentrés sur les problèmes d'allocation des tâches qui ont des restrictions entre les tâches. Hoplites [240, 296] est un système basé sur le marché qui applique à la fois une coordination passive et active. Les tâches sont effectuées plus rapidement grâce à une coordination passive, ce qui réduit l'utilisation de la communication. La coordination active est utilisée pour permettre aux agents d'influer activement sur les enchères des autres agents. Des algorithmes basés sur le marché pour le routage des robots dans les espaces ouverts sont présentés dans Mosteo et al. [297]. Guerrero et Oliver [298] ont présenté des approches de marché pour trouver un nombre optimal de robots nécessaires pour résoudre des tâches spécifiques. Nanjanath et Gini [299] ont présenté une variante de CNP pour résoudre les problèmes d'allocation des tâches. Leur système permettait de réaffecter les tâches pour faire face à des environnements dynamiques. Zlot et Stentz [284] ont abordé l'allocation basée sur le marché

de tâches complexes avec des phases de décomposition et d'allocation des tâches considérées ensemble au lieu d'être séparées comme d'habitude.

Les approches fondées sur le marché saisissent les forces respectives des approches distribuées et centralisées [217]. Ils peuvent répartir une grande partie de la planification et de l'exécution sur l'équipe et ainsi conserver les avantages des approches distribuées, notamment la robustesse, la flexibilité et la rapidité. Ils ont également des éléments de systèmes centralisés pour produire de meilleures solutions : les enchères rassemblent de manière concise des informations sur l'équipe et distribuent les ressources dans un contexte conscient de l'équipe. Cependant, si les coûts de communication sont trop élevés dans le processus d'allocation des tâches, une fois qu'il y a des échecs dans la communication de l'agent, les performances se dégraderont sensiblement [300], de sorte que ces méthodes conviennent à l'allocation de tâches à petite et moyenne échelle. Le problème d'allocation des tâches peut être résolu par divers algorithmes et méta heuristiques.

IV.4. Algorithmes pour l'attribution de tâches multi-capteurs

IV.4.1. Algorithme des abeilles distribuées (Distributed Bees Algorithm ou DBA)

Dans cet algorithme, chaque capteur est représenté comme une "abeille". Si on prend le cas de l'allocation de tâche pour des robots/cobots distribués dans un espace, lorsqu'un robot (capteur) reçoit des informations sur les cibles (tâches), il calcule les utilitaires par rapport à ces cibles. L'utilité dépend de la valeur de qualité de la cible et du coût associé, c'est-à-dire de la distance du robot par rapport à la cible. C'est le concept de base derrière le DBA qui a été introduit dans [301], il est décrit en plus de détails ici.

- 1) Coûts : Le coût d'une cible i pour le robot k est calculé comme la distance euclidienne entre le robot et la cible et si on néglige les obstacles entre eux dans une arène 2D :

$$d_i^k = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} \quad (\text{Equ.IV.1})$$

Où (x_i, y_i) et (x_k, y_k) représentent respectivement les coordonnées de la cible (tâche) et du robot (capteur/ nœud agent) dans l'arène (un espace). Cependant, pour calculer l'utilité, on utilise la visibilité de la cible définie comme la valeur réciproque de la distance :

$$\eta_i^k = 1/d_i^k \quad (\text{Equ.IV.2})$$

- 2) Qualités : La qualité est une valeur scalaire qui représente la priorité ou la complexité de la cible. Les qualités normalisées sont calculées comme des fractions de la somme des qualités de toutes les cibles disponibles :

$$q_i = \frac{Q_i}{\sum_{j=1}^M Q_j} \quad (\text{Equ.IV.3})$$

Où Q_i est une qualité de la cible i . Dans les scénarios du monde réel, la qualité d'une région d'intérêt est une valeur estimée résultant de lectures de capteurs ou de connaissances acquises antérieurement. Dans un problème d'application de la loi, par exemple, le type de la tâche détermine sa priorité. Une plainte pour bruit aura une priorité inférieure à un vol. Un meurtre aura une priorité plus élevée qu'un vol. Dans un problème de sécurité du réseau d'approvisionnement, une porte déverrouillée aura une priorité inférieure à une cargaison manquante, mais une cargaison manquante aura une priorité inférieure à un incendie dans un centre logistique central.

- 3) Délai : Chaque tâche a une limite de temps, ou une durée du temps, le délai appelé Δ_i est fonction de la priorité (qualité), où Δ_i diminue avec le temps jusqu'à atteindre une valeur de 0, ce qui indique que la tâche i n'est plus pertinente. Une fois qu'un capteur est affecté à une tâche, il surveille cette tâche particulière (avec un V^k_i de performance). Une tâche est considérée comme détectée une fois que des capteurs lui sont affectés.
- 4) Utilitaires de calcul : L'utilité d'un robot tel que proposé en [208, 241] dépend à la fois du coût et de la qualité de la cible choisie. On peut définir l'utilité comme une probabilité que le robot k soit alloué à la cible (tâche) i , et dépend à la fois de la priorité (complexité) et du coût (la distance tâche/capteur). On prend la visibilité moyenne d'une tâche η^k_{moy} par rapport aux qualités des cibles telle que :

$$\eta^k_{\text{moy}} = E[\eta^k_i(q_i)] = \sum_{i=1}^M q_i \eta^k_i \quad (\text{Equ.IV.4})$$

Si on prend en compte une notion d'importance de tâche/visibilité on fait injecter deux paramètres α et β dont $(q_i)^\alpha$ et $(\eta^k_i)^\beta$ comme des paramètres de contrôle d'importance pour biaiser le mécanisme de décision vers la qualité de la solution ou son coût (ou bien sa visibilité), respectivement ($\alpha, \beta > 0$; $\alpha, \beta \in \mathbb{R}$), alors :

$$\eta_{\text{moy}}^k = \sum_{i=1}^M (q_i)^\alpha (\eta_i^k)^\beta \quad (\text{Equ.IV.5})$$

Donc et pour $\Delta_i > 0$, l'utilité est calculée comme suit :

$$p_i^k = \frac{(q_i)^\alpha (\eta_i^k)^\beta}{\eta_{\text{moy}}^k} \quad (\text{Equ.IV.6})$$

À partir de (Equ.IV.6), il est facile de montrer que :

$$\sum_{i=1}^M p_i^k = 1 \quad (\text{Equ.IV.7})$$

- 5) **Prise de décision** : Le mécanisme de prise de décision sous-jacent de l'algorithme DBA adopte la règle de la roulette, également connue sous le nom de règle de sélection de roue. Autrement dit, chaque cible a une probabilité associée avec laquelle elle est choisie parmi un ensemble de cibles disponibles. Une fois toutes les probabilités calculées comme en (Equ.IV.6), le robot choisira une cible en « faisant tourner la roue ». Il convient de noter que la distribution des robots résultants dépend de leur distribution initiale dans l'arène, c'est-à-dire de leurs distances par rapport à chaque cible avant l'attribution de la cible. Par conséquent, les utilitaires des robots diffèrent par rapport à la même cible si leurs distances par rapport à cette cible ne sont pas égales. Étant donné qu'un utilitaire de robots combinés ne peut pas être calculé en raison de la nature distribuée de l'algorithme proposé, la qualité des cibles est utilisée comme seule mesure de la distribution attendue des robots.

IV.4.2. Algorithme des abeilles distribuées hétérogènes

Les chercheurs ont proposé un algorithme d'abeilles distribuées (DBA) qui convient à la mise en œuvre dans un système multi-capteur où les agents coopérants interagissent en échangeant des informations disponibles localement, de sorte que l'objectif global soit obtenu plus efficacement qu'il ne le serait par des robots qui exécutent des tâches individuellement. Cet algorithme d'intelligence en essaim a été amélioré pour contrôler une allocation de réseau de capteurs hétérogène.

La fonction DBA de l'utilité des robots a été modifiée (HDBA) par [205] pour le cas de capteurs hétérogènes aux performances différentes. Cette modification est supposée améliorer les performances du système car elle peut corrélérer la fonction d'utilité des robots avec la valeur de leurs performances.

Afin de définir HDBA pour des robots hétérogènes, la valeur d'utilité d'une nouvelle tâche (V_i^k) est définie en fonction des performances du robot sur cette tâche. Lorsqu'un capteur reçoit des informations sur les tâches disponibles, il calcule ses performances pour cette tâche. La fonction utilitaire du robot est mise à jour en conséquence et dépend de la priorité de la tâche, de la distance (ou la visibilité) par rapport à la tâche et des performances du robot sur cette tâche (pour $\Delta_i > 0$) :

$$p_i^k = \frac{(q_i)^\alpha (\eta_i^k)^\beta (V_i^k)^\gamma}{\sum_{i=1}^M (q_i)^\alpha (\eta_i^k)^\beta (V_i^k)^\gamma} \quad (\text{Equ.IV.8})$$

Où γ est un paramètre de contrôle qui biaise l'importance des performances du robot et V_i^k est la performance du robot k sur la tâche i .

Le mécanisme de prise de décision HDBA applique la même règle de sélection de roue utilisée dans DBA pour choisir parmi un ensemble de tâches disponibles. Les tâches arrivant à des moments différents déclenchent une nouvelle itération décisionnelle de l'algorithme.

Considérant, deux tâches différentes sont arrivées au robot k à des moments différents. La priorité normalisée (la qualité q_i^k) de la tâche 1 est donnée par q_1^k et de la tâche 2 par q_2^k . La distance euclidienne des robots de la tâche 1 et de la tâche 2 est respectivement d_1^k et d_2^k . Les performances des capteurs sur la tâche 1 et la tâche 2 sont respectivement V_1^k et V_2^k . Les probabilités du $k^{\text{ème}}$ robot à affecter à la tâche 1 et à la tâche 2 sont respectivement $p_1^{k^{\text{ème}}}$ et $p_2^{k^{\text{ème}}}$. A l'arrivée de la première tâche, HDBA est exécuté pour calculer les probabilités d'allocation de chaque robot. Puisqu'il n'y a pas d'autres tâches, les robots sont affectés à la tâche qui se trouve dans leur plage d'allocation, et ils coopèrent pour mener à bien cette tâche. A l'arrivée de la deuxième tâche, une autre itération décisionnelle de HDBA est déclenchée pour calculer les nouvelles probabilités d'allocation des robots. Sur la base des probabilités calculées, HDBA converge vers une nouvelle allocation. Dans cette attribution, des robots supplémentaires sont attribués à la deuxième tâche et certains robots qui ont été attribués à la première tâche sont réaffectés à la deuxième tâche, de sorte que le temps global d'achèvement de la tâche est minimisé. HDBA est exécuté jusqu'à ce que la condition de terminaison soit remplie.

Le DBA / HDBA, similaire à d'autres algorithmes d'intelligence en essaim, comprend une population d'agents simples interagissant localement les uns avec les autres et avec leur environnement. Les agents font une construction itérative de solutions pour rechercher l'espace

des solutions en tenant compte des solutions précédentes pour optimiser leur recherche. Il s'agit d'un algorithme collaboratif probabiliste qui utilise une distribution de probabilité pour effectuer la transition entre chaque itération. Le caractère probabiliste de cet algorithme lui permet d'éviter l'optimum local en choisissant des solutions parfois pires en utilisant la règle de sélection de roue. Accepter les pires solutions est une propriété fondamentale de la méta-heuristique car elle permet une recherche plus approfondie de la solution optimale globale.

IV.4.3. Système d'abeille

L'algorithme Bee System (BS) utilisé dans celui de Lučić et Teodorovic [302] pour un problème de voyageur de commerce est décrit. Cet algorithme est basé sur le comportement de recherche de nourriture d'une colonie artificielle d'abeilles pour allouer de manière optimale des capteurs aux tâches. Lorsqu'une nouvelle tâche est introduite, l'algorithme est capable de remplacer l'allocation obtenue par une nouvelle allocation améliorée.

$$P_i^k = \frac{e^{\alpha q_i - \beta d_i^k}}{\sum_{j=1}^M e^{\alpha q_j - \beta d_j^k}} \quad \text{pour } \Delta_i > 0 \quad (\text{Equ.IV.9})$$

Où α et β sont des paramètres de contrôle qui biaisent l'importance de la priorité et de la distance, respectivement ($\alpha, \beta > 0$; $\alpha, \beta \in \mathbb{R}$).

Le robot donc k_0 de meilleur $P_{i_0}^{k_0}$ va être orienté vers la tâche i_0 .

IV.4.4. Optimisation des colonies de fourmis

Un algorithme d'essaim de colonie de fourmis appelé Ant Colony Optimization (ACO) a été développé par Dorigo et al. [303]. Dans cette méthode, chaque fourmi est un agent comme si un robot qui choisit des points destinations avec une probabilité qui est fonction de la distance ou la visibilité vers la destination dans un endroit et de la quantité ou clarté de traces sur les bords du chemin. Pour forcer la fourmi à faire des cycles de visites légales, les transitions vers des points déjà visités sont interdites jusqu'à ce qu'un cycle soit terminé (ceci est contrôlé par une liste des interdits) ; lorsqu'il termine un cycle, il dépose une substance appelée traînée sur chaque coin (bord) (i,j) visité, où i est l'endroit qui renferme un ensemble de fourmis b_i et dans une durée de cycle dont $i=1,2,\dots,n$ et la somme $\sum_{i=1}^n b_i = m$ représente le nombre de toutes les fourmis, et j représente le bord référencé.

$$v_{ij}(t+n) = \rho \times v_{ij}(t) + \Delta v_{ij}, \quad \text{si } v_{ij} \geq v_0 \quad (\text{Equ.IV.10})$$

Où $v_{ij}(t)$ est l'intensité de la traînée de la fourmi dans (i,j) au temps t , chaque fourmi à l'instant t choisit l'endroit suivant, où elle sera à l'instant $t+1$. Par conséquent, si nous appelons une itération de l'algorithme ACO les m déplacements effectués par toutes les fourmis de nombre m dans l'intervalle $(t, t+1)$, alors toutes les n itérations de l'algorithme (cycle) chaque fourmi a effectué un cycle dans tous les endroit i . ρ est un coefficient tel que $(1 - \rho)$ représente l'évaporation de traînée entre l'instant t et $t + n$, v_0 est un seuil.

$$\Delta v_{ij} = \sum_{k=1}^m \Delta v_{ij}^k \quad (\text{Equ.IV.11})$$

Où Δv_{ij}^k est la quantité par unité de traînée déposée sur le coin (i,j) par la k -ième fourmi entre l'instant t et $t + n$; il est donné par $\Delta v_{ij}^k = Q/L_k$ si la k -ième fourmi utilise le coin (i,j) entre t et $t+n$ et $\Delta v_{ij}^k = 0$ ailleurs, où Q est une constante et L_k est la longueur du cycle de la k -ième fourmi. La liste des interdits enregistre les bords déjà visités jusqu'à l'instant t et interdit à la fourmi de les visiter à nouveau. Lorsqu'un tour est terminé, la liste des interdits est utilisée pour calculer la solution actuelle de la fourmi (c'est-à-dire l'ensemble de bords atteint).

La probabilité de transition dans l'endroit i pour la k -ième fourmi est donnée par :

$$P_{ij}^k = \begin{cases} \frac{(v_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{k \in J} (v_{ik}(t))^\alpha \cdot (\eta_{ik})^\beta} & \text{si } j \in J \\ 0 & \text{ailleurs} \end{cases} \quad (\text{Equ.IV.12})$$

Où la visibilité η_{ij} est la performance collective des bords sélectionnés S , J est l'ensemble des coins ne figurant pas dans la liste des interdits, α et β sont des paramètres qui contrôlent l'importance relative de la piste par rapport à la visibilité.

IV.4.5. Algorithme basé sur le marché (l'enchère)

Un algorithme basé sur le marché qui utilise des offres d'agents dans des enchères pour des tâches de détection distribuées est présenté. Les agents se font concurrence pour gagner des tâches en plaçant des offres dans des enchères pour produire des allocations efficaces. Chaque agent distribué calcule un coût pour terminer une tâche et diffuse l'offre pour cette tâche. L'agent du commissaire-priseur décide de la meilleure offre disponible, et le soumissionnaire retenu tente d'exécuter la tâche (Fig.IV.2). Dans cet algorithme, l'offre du robot k à la tâche i est définie comme (pour $\Delta_i > 0$) :

$$\text{Bid}_i^k = g_i^k + \delta(\eta_i^k v_i^k - g_i^k) \quad (\text{Equ.IV.13})$$

Où g_i^k sert de prix de réservation de la tâche i , δ est un paramètre de contrôle avec des valeurs comprises entre 0 et 1, et η_i^k représente la visibilité. Le capteur k est alloué à une tâche s s'il maximise Equ.IV.14.

$$\text{T\^ache} - \text{allou\^ee} = \max(\text{Bid}_i^k) \quad (\text{Equ.IV.14})$$

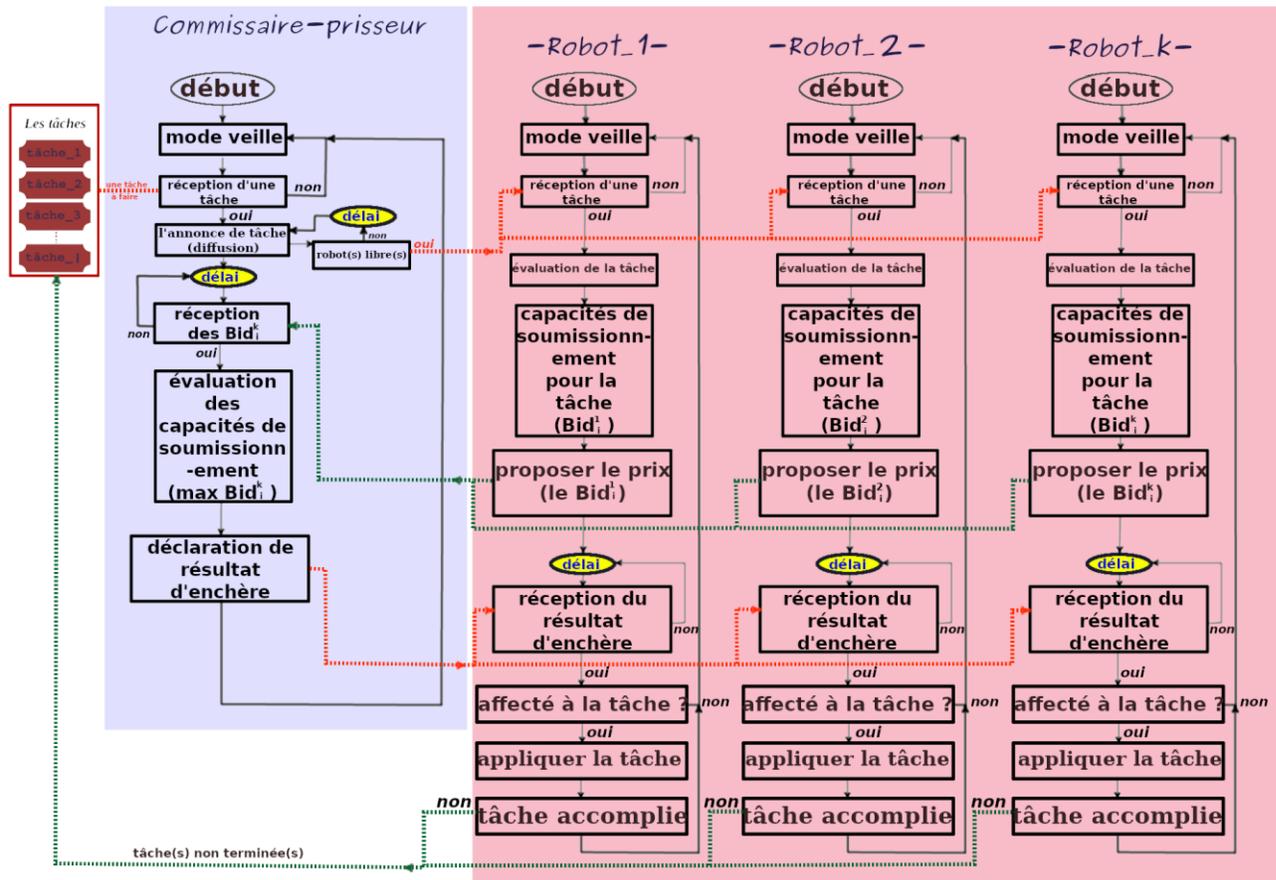


Fig.IV.2 : Diagramme de flux de travail pour l'algorithme basé sur le marché (inspiré par Tkach et al. [304])

On peut distinguer des différentes contributions d'algorithmes basés sur les enchères tels que énoncés dans [305]. Ils décrivent en comparant des algorithmes des différentes enchères, notamment : les enchères séquentielles, parallèles, G-Prim, répétées parallèles, répétées G-Prim et combinatoires. Ils proposent des algorithmes qui supposent un objectif de maximisation du profit. Chaque message est soit diffusé par le commissaire-priseur, soit envoyé une fois par chaque non-commissaire. Une diffusion équivaut à envoyer le même message une fois à chaque agent ; les auteurs [305] supposent que le succès ou l'échec d'un message diffusé atteignant différents agents

est indépendant. Ils supposent que les messages envoyés par le commissaire-priseur à lui-même ne sont jamais abandonnés. Dans une enchère séquentielle, il y a m tours d'enchères différents, un pour chacun des m articles, et chaque message nécessite un espace $O(1)$. Dans une enchère parallèle, il y a un seul tour (dans lequel chacun des m articles est simultanément enchère), et les messages entre agents (et à n'importe quel gestionnaire) nécessitent l'espace $O(m)$. Dans l'algorithme G-Prim, le gestionnaire annonce tous les articles invendus à chaque tour et l'agent enchérit pour son premier choix à chaque tour. Cela oblige les gestionnaires à envoyer des messages de taille $O(m)$, tandis que les agents envoient des messages de taille $O(1)$. Dans la variante répétée de l'enchère parallèle, chaque agent gagne un seul objet à chaque tour et des tours $O(m/n)$ sont nécessaires. À chaque tour, le commissaire-priseur et les agents envoient des messages de taille $O(n)$. L'enchère répétée G-Prim est similaire, sauf que le commissaire-priseur annonce tous les articles invendus à chaque tour en utilisant des messages de taille $O(m)$ (alors l'agent enchérit pour leurs n premiers choix des articles restants à chaque tour). Dans une vente aux enchères combinatoire, il y a un seul tour (dans lequel chacune des $O(m!)$ combinaisons possibles d'articles est proposée par chaque agent), et les messages entre agents (et à tout gestionnaire) nécessitent l'espace $O(m!)$ dans le pire cas.

IV.4.6. Fisher Market Clearing Task Allocation (FMC_TA)

Un algorithme pour une compensation du marché des pêcheurs pour l'allocation des tâches avec d'agent hétérogène (FMC_TAH+) calcule une allocation en modélisant les agents/robots en tant qu'acheteurs, les tâches en tant que biens et les utilités en tant que préférences [301]. Les tâches individuelles, particulièrement les plus importantes, peuvent être réparties entre plusieurs robots qui partagent la charge de travail. FMC_TAH+ planifie l'allocation sous contraintes inter-tâches et inter-agents. Les tâches attribuées à chaque agent robot sont classées par ordre d'importance. Les séquences de tâches sont ensuite modifiées en fonction des contraintes inter-agents. Plus précisément, étant donné que les tâches partagées ne peuvent pas être démarrées tant que tous les agents affectés ne sont pas arrivés, les agents ne bénéficient pas d'arriver plus tôt que le dernier agent. Pour chaque agent programmé pour arriver tôt à une tâche partagée, la prochaine tâche est programmée avant la tâche partagée si ce déplacement ne retarde pas l'exécution de la tâche partagée. Les étapes de l'algorithme FMC_TAH+ sont :

1. Générer une instance de marché pêcheur (Market Fisher) ;

2. Trouver une solution de compensation de marché pêcheur et acquérir une allocation de tâches aux agents robots ;
3. Ordonnez les tâches attribuées à chaque agent robots.

Une solution de compensation de marché est un vecteur de prix Φ spécifiant un prix Φ_i^k pour chaque bien (tâche) i qui permet à chaque acheteur (robot) k de dépenser tout son argent pour des biens qui maximisent le rapport qualité/prix (Q_i^k/Φ_i^k) tandis que tous les biens sur le marché sont vendus [301].

IV.4.7. Algorithme glouton

Un algorithme glouton pour un problème d'observation multi-tâches avec la messagerie de diffusion est présenté. Cet algorithme est configuré pour effectuer une allocation de capteur basée sur la meilleure allocation possible de chaque capteur individuel à une tâche qui maximise le rapport performances sur coût (V_i^k / d_i^k), où V_i^k est la performance du k -ème capteur sur la i -ème tâche et d_i^k est la distance euclidienne entre le capteur et la tâche.

$$T_i^k = \max_{i \in Z} (V_i^k / d_i^k) = \max_{i \in Z} (V_i^k \eta_i^k) \quad \text{pour } \Delta_i > 0 \quad (\text{Equ.IV.15})$$

Où la tâche i est la tâche choisie du k -ème capteur parmi toutes les tâches possibles pour l'affectation, et Z est le groupe de tâches dans la k -ème page de capteur sur M tâches disponibles.

Daniel Tihanyi et al. [306] introduisent deux heuristiques gourmandes, un algorithme glouton direct (Forward Greedy Algorithm) et glouton inverse (Reverse Greedy Algorithm). D'un point de vue application, ils peuvent réussir de concevoir un système théorique (idée) sur un plan 2D à base des robots mobiles en collaborant afin de survivre ou sauver des cibles (personnes/ objets...) dans un scénario de sauvetage d'urgence pour une sécurité contre les incertitudes dynamiques, telles que le déplacement d'obstacles et l'évolution des dangers. D'un point de vue technique en présence de ces algorithmes mettent à jour de manière itérative l'allocation des tâches en ajoutant ou en supprimant une paire tâche-robot à chaque étape. De plus, grâce à sa forme multiplicative, l'objectif de [306] peut être séparé entre les robots et mis en œuvre de manière basée sur des enchères.

Ils ont divisé leur travail en deux parties, une planification de chemin comme étape de bas niveau (optimisation des politiques de contrôle de chaque robot individuellement pour les cibles assignées : définir un processus de Markov stochastique combinant le modèle du mouvement du

robot, sa connaissance de la propagation du risque et sa progression vers l'accomplissement de sa mission). Et la deuxième étape c'est l'allocation de tâches comme étape de haut niveau, et celle, est la partie nécessaire dans cette section.

L'étape de haut niveau attribue les cibles aux robots pour maximiser un objectif collectif. Cette partie s'appuie sur les solutions obtenues par l'étape bas niveau dans [306].

Supposant qu'il suffit qu'une cible soit visitée une fois. Ainsi, une allocation de tâche valide assigne chaque cible à exactement un robot qui va partitionner l'ensemble T qui représente toutes les tâches disponibles en $\{ T^k \}$ pour $k \in R$ l'ensemble de tous les robots disponibles. Pour être plus précis, T^k inclus dans T pour tout $k \in R$, $T^k \cap T^{k'} = \emptyset$ pour toute paire $k, k' \in R$ où k est différent de k' et l'union de tout T^k pour $k \in R$ donne T . Chaque T^k représente le sous-ensemble affecté au robot k .

L'objectif était de trouver la répartition des tâches qui maximise la probabilité de réussite du groupe en montrant que le succès multiplicatif du groupe est une borne inférieure de la probabilité réelle de succès du groupe dans des conditions modérées, c'est-à-dire la probabilité que tous les robots terminent leurs missions en toute sécurité. Ainsi, ils introduisent le succès multiplicatif du groupe comme une approximation de la fonction objectif. Il est défini comme $F(\{ T^k \}_{k \in R}) = \prod_{k \in R} f_k(T^k)$, où les valeurs de $f_k(T^k)$ sont obtenues en résolvant le problème de planification de trajectoire mono-robot introduit dans l'étape de niveau bas pour chaque $k \in R$.

Le problème de répartition des tâches pour maximiser la probabilité de sécurité peut être formulé comme suit :

$$F^* = \text{Best}_{\{T^k\}_{k \in R}} \prod_{k \in R} [f_k(T^k)] \quad (\text{Equ.IV.16})$$

Le problème ci-dessus est une instance de partitionnement d'ensembles et est NP-difficile. Dans la section suivante, nous proposons deux variantes des algorithmes gloutonnes pour permettre des implémentations distribuées efficaces de l'allocation des tâches pour maximiser l'achèvement de la mission en toute sécurité.

Pour les deux heuristiques gloutonnes, la forward et la reverse, d'un point de vue pratique, ces algorithmes mettent à jour de manière itérative l'allocation des tâches en ajoutant ou en supprimant une paire tâche-robot à chaque étape. De plus, grâce à sa forme multiplicative, l'objectif peut être découplé parmi les robots et mis en œuvre de manière enchère. A chaque itération, chaque robot peut soumettre une offre en fonction de son objectif individuel f_k . Les offres sont ensuite collectées

par une unité centrale afin qu'une décision basée sur la fonction objective de groupe F puisse être prise.

L'algorithme glouton direct ou forward est initialisé sans aucune tâche allouée à aucun des robots. Il met ensuite à jour cette allocation de manière itérative en choisissant la paire robots/tâches avec le meilleur gain d'optimalité jusqu'à ce que chaque tâche soit allouée à un robot. Le calcul des étapes d'itération est réparti entre les robots.

Algorithme IV.1 : L'algorithme glouton direct

Début :
 Initialisation :
 $T^0_k = \emptyset, f^0_k = f_k(\emptyset), J_0 = T, R_0 = R$ #pour tout $k \in R$
Pour $i = 1, 2, 3, \dots, |T|$ **faire :**
 Pour $k \in R_{i-1}$ **faire :**
 # les offres soumis par chaque robot
 $t^{k_i} \leftarrow t \mid \max(f_k)$
 $\delta^{k_i} \leftarrow \delta \mid f_k(t^{k_i})$
 Fin
 $\{(t^{k_i}, \delta^{k_i})\}_{k \in R} = \{(t^{k_i}, \delta^{k_i})\}_{k \in R} + \{(t^{k_{i-1}}, \delta^{k_{i-1}})\}_{\forall k \in R_{i-1} = \emptyset}$ #collecter toutes les offres
 $k_i \leftarrow k \mid F^*$ #le robot de meilleur gain d'optimalité par rapport à l'objectif collectif
 $T^{k_i} \leftarrow T^{k_i} \cup t^{k_i}$ pour $k=k_i$; $T^{k_{i-1}}$ pour $k \neq k_i$ #écraser la nouvelle tâche allouée et laisser
 $f^{k_i} \leftarrow f^{k_i} - \delta^{k_i}$ pour $k=k_i$; $f^{k_{i-1}}$ pour $k \neq k_i$ les autres
 $R_i \leftarrow \{k \mid t^{k_i} = t^{k_i}\}_{k \neq k_i}$ si $|\{k \mid t^{k_i} \in T^{k_i}\}| > 1$; $R \setminus \{k_i\}$ ailleurs
 $J_i \leftarrow J_{i-1} \setminus t^{k_i}$ # retirer la tâche allouée à k_i
Fin
Fin.

L'algorithme glouton direct se déroule comme suit. Tout d'abord, définissant les variables suivantes pour chaque étape i : $\{T^k_i\}_{k \in R}$ désigne l'allocation de tâches actuelle, et $\{f^k_i\}_{k \in R}$ stocke les valeurs de fonction d'évaluation pour chaque robot k .

De plus, J_i est l'ensemble des tâches qui n'ont pas encore été allouées et R_i est l'ensemble des robots qui doivent mettre à jour leurs offres (Bids) à l'étape suivante. Initialement, aucune tâche n'est affectée, donc $T^k_0 = \emptyset$ pour tout $k \in R$. À chaque étape, exactement une tâche est allouée, nous avons donc besoin de $|T|$ étapes, le nombre des tâches, pour terminer l'algorithme. À chaque itération i , tous les robots $k \in R$ soumettent une offre, qui se compose de la paire (t^k_i, δ^k_i) . Chaque robot k choisit la tâche t^k_i dans la liste des tâches non allouées J_{i-1} , de telle sorte qu'il obtienne le meilleur gain d'optimalité δ^k_i par rapport à la fonction objectif individuelle f_k . Après avoir collecté

toutes les offres, on choisit le robot k_i qui génère le meilleur gain d'optimalité par rapport à l'objectif collectif : le succès multiplicatif du groupe F . Grâce à la formulation basée sur les enchères, on peut choisir efficacement le couple tâche-robot avec le meilleur gain collectif. Ensuite, on définit simplement les valeurs de f_i^k, T_i^k pour tout $k \in R$ et R_i, J_i en fonction du choix résultant dans la répartition des tâches (seuls les robots choisissant la même tâche que k_i , c'est-à-dire k tel que $t_i^k = t_i^{k_i}$, doivent mettre à jour leurs offres à l'itération suivante. Les autres robots soumettent simplement leurs offres de l'itération précédente. La variable R_i est initialisée avec $R_0=R$, puisque dans la première itération tous les robots doivent calculer leurs offres).

Pour une garantie de performance en fonction de la valeur $F(\{\emptyset\} k \in R)$, l'allocation vide initiale. Cela pourrait potentiellement prendre une valeur élevée, par exemple 1, ce qui pourrait détériorer la garantie de performance. Pour résoudre ce problème, Daniel Tihanyi et al. [306] introduisent ensuite l'algorithme glouton inverse.

L'algorithme glouton inverse est initialisé avec toutes les tâches attribuées à chaque robot simultanément. Pour cette raison, sa garantie de performance sera plutôt fonction de la valeur $F(\{T\} k \in R)$. Cet algorithme met à jour de manière itérative son allocation provisoire en supprimant des tâches des robots. À chaque étape, la paire tâche-robot causant la plus grande perte d'optimalité est supprimée. Il converge lorsque chaque tâche est attribuée à exactement un robot. Comme c'est le cas pour le forward glouton, le calcul des pas d'itération est réparti entre les robots. Étant donné que la mise en œuvre gloutonne inverse utilise des principes similaires à ceux trouvés dans le forward glouton.

IV.4.8. Algorithme génétique

Un algorithme génétique (AG) est un algorithme évolutif utilisé pour résoudre des problèmes d'optimisation avec une complexité non polynomiale. L'évolution commence généralement à partir d'une population d'individus générés aléatoirement et constituée de solutions candidates légitimes, et est un processus itératif, la population de chaque itération étant appelée une génération. À chaque génération, l'aptitude de chaque individu de la population est évaluée ; le fitness est généralement la valeur de la fonction objective dans le problème d'optimisation en cours de résolution. Les individus les plus en forme sont sélectionnés de manière stochastique dans la population actuelle, et le génome de chaque individu est modifié par croisement et mutation pour former une nouvelle génération. La nouvelle génération de solutions candidates est ensuite utilisée dans l'itération suivante de l'algorithme. Généralement, l'algorithme se termine lorsque soit un

nombre maximum de générations a été produit, soit un niveau satisfaisant de la fonction objectif a été atteint.

Algorithme IV.2 : L'algorithme génétique pour l'allocation de tâches

```
Début :  
Initialisation :  
  G = Générer des génotypes aléatoires représentant des solutions légitimes  
  du robot aux tâches ; #l'ensemble de robot-tache avec des paramètres tels que coût et  
  performance...  
  
  Fki = 0; # valeur minimale de la fitness pour chaque robot-tache  
  Seuil = Cte ; #niveau satisfaisant du fitness  
Pour  $\forall$  chromosome  $\in G$  faire :  
  Calculer la fitness Fki;  
  Tki ← tki / max(Fki); #affectation de tâches soit selon des paramètres précédentes ou d'une manière  
Fin aléatoire  
Tant que  $F^{k_i} < \text{seuil}$  faire :  
  # Générer la nouvelle population  
  Sélectionnez les parents ;  
  Appliquer le croisement en recombinant des paires de parents ;  
  Appliquer la mutation à la progéniture en modifiant une allocation dans un  
  génotype ;  
  # comparaison et optimisation  
  Évaluer de nouvelles solutions en calculant des fonctions de fitness Fk'i;  
  Si  $F^{k_i} < F^{k'_i}$  faire :  
    Fki ← Fk'i; # Stocker la meilleure fonction de fitness pour chaque robot vers une tâche  
    Tki ← tki / max(Fki); # Stocker le chromosome qui obtient la meilleure fonction de fitness pour chaque robot vers  
  Fin une tâche  
Fin  
Prendre le chromosome stocké ;  
Fin.
```

IV.4.9. Recuit simulé (Simulated Annealing)

Le recuit simulé (SA) est une méta-heuristique inspirée du recuit physique en métallurgie. Il utilise la fonction objective qui maximise implicitement l'utilité de l'équipe. SA explore l'espace des solutions en utilisant des transitions aléatoires pour générer de nouvelles solutions et en réutilisant la valeur de température pour converger. SA permet des transitions occasionnelles menant à des solutions pires, mais cela aide à empêcher la recherche de rester bloquée dans les optima locaux.

Algorithme IV.3 : L'algorithme Simulated Annealing

```

Début :
Initialisation :
  Sol = créer une solution initiale ;
  Soli = ∅ ;
  Iter = spécifier le nombre d'itérations ;
  t = initialiser la température ; # est une valeur de température théoriquement
Tant que cost(Sol) != cost(Soli) faire : utilisée
  Pour i=1 vers Iter faire :
    Generer une solution aléatoirement / Soli != Sol ;
    Si cost(Sol) > cost(Soli) faire :
      Sol = Soli ;
    D'autre si exp( (cost(Sol) - cost(Soli))/t ) > ( random [0.1[ ] ) :
      Sol = Soli ;
  Fin
Fin
  t = réduire la température ;
Fin
  Prendre la solution Sol
Fin.

```

IV.5. Discussion

L'allocation de tâche est une partie avec laquelle elle représente un thème de l'époque qui renferme une large base de données de recherches et publications dans le domaine. Si on parle de l'heuristique et de l'optimisation, il y a beaucoup d'autres algorithmes comme l'algorithme de Newton-Raphson, Steepest Algorithm, Bracketing Algorithms for Root Finding, etc, mais toujours le problème des optimaux locaux dépose un obstacle qui met en stade de penser à des solutions amélioré pour pénétre ce problème, c'est ce qui a conduit les chercheurs dans le domaine à créer un nouveau terme appelé un méta-heuristique qui ne dépends pas sur des prés acquis à la volé de l'algorithme, c'est ce dont on a discuté dans cette section en mentionnant quelques algorithmes et beaucoup d'autres qui sont utilisés où peuvent être utilisés pour une gestion de processus d'allocation de tâches à savoir Particle Swarm Optimisation, Biogeography-Based Optimization, Differential Evolution Algorithm, Artificial Bee Colony, etc. Une nouvelle approche qui peut encore faire un bond pour le mieux par l'utilisation de l'intelligence artificielle et que peut-on en faire dans ce que l'en appelle hyper-heuristique ou la recherche heuristique pour l'heuristique pour la résolution des problèmes et sortir par au moins une solution (optimum global).

Le choix d'un algorithme permis d'autres dépend en premier lieu du type d'application où les robots/cobots doivent collaborer dedans. Si le critère, comme exemple, distance (coût) est majeur donc on peut adopter la DBH, HDBA ou même la SA pour augmentation du β le paramètre de contrôle d'importance, ou bien si le support de traitement peu puissant, l'utilisation de l'algorithme glouton est un choix acceptable. Si on a un scénario de visite, d'exploration, de recherche sur des cible l'algorithme ACO peut aider avec efficacité. Si on parle sur des problématiques dans des processus stochastique et difficile de le déterminer d'une manière sûre, le GA ou le SA est un choix rationnel. Pour des taches spécifiée ou prédéfinies et ont des paramètres critères pour comparaison entre eux avec des capacités de robots/cobots donc on peut choisir soit l'algorithme basé sur les enchères, comme exemple, l'algorithme glouton aussi, et pour une distributions et répartition en partageant les taches le FMC_TA peut être un choix pertinent.

IV.6. Système d'exploitation en temps réel dans la MRTA

L'allocation de tâches pour multirobot (MRTA) est un domaine complexe de la robotique qui traite de la distribution efficace de tâches entre plusieurs robots dans un environnement collaboratif. Un des aspects clés pour assurer la performance, la réactivité, et la coordination des robots dans un système MRTA est l'utilisation d'un système d'exploitation en temps réel (RTOS).

Un système d'exploitation en temps réel (RTOS pour Real Time Operating System) est un type de système d'exploitation conçu pour gérer les ressources informatiques afin de garantir que les tâches critiques sont exécutées dans des délais stricts et prévisibles. Contrairement aux systèmes d'exploitation généraux qui optimisent pour le débit global, un RTOS se concentre sur la minimisation de la latence et sur le respect des contraintes temporelles.

IV.6.1. Avantages d'un RTOS dans un système MRTA

Dans un environnement multirobot, le respect des délais est crucial pour assurer une coordination efficace et éviter les conflits. Un RTOS offre un comportement déterministe, garantissant que les tâches critiques sont exécutées à des moments précis. Pour la priorisation des Tâches, les RTOS permettent de définir des priorités pour les tâches, assurant que les tâches les plus importantes ou les plus urgentes sont traitées en premier. Cela est essentiel dans un système MRTA où certaines tâches peuvent avoir des implications de sécurité ou de coordination critiques. Un RTOS est conçu pour gérer efficacement les interruptions, ce qui permet une réponse rapide aux événements externes, tels que les capteurs des robots détectant des obstacles ou recevant des nouvelles instructions. Pour la concurrence et la synchronisation, les RTOS fournissent des

mécanismes robustes pour la gestion de la concurrence et la synchronisation des tâches, réduisant les risques de conditions de concurrence et de blocages, qui sont fréquents dans les systèmes multirobots.

IV.6.2. Intégration d'un RTOS dans MRTA

Dans un système MRTA, les tâches doivent être allouées de manière dynamique en fonction de la disponibilité des robots, des priorités des tâches, et des contraintes temporelles. Un RTOS peut faciliter cette planification en assurant que les tâches sont distribuées et exécutées en respectant les contraintes temporelles et les priorités définies. Les RTOS offrent aussi des protocoles de communication inter-tâches efficaces, permettant une meilleure coordination entre les robots. Par exemple, un RTOS peut garantir que les messages critiques entre robots sont transmis et reçus sans délai excessif. De plus, ils gèrent les ressources matérielles et logicielles de manière optimisée, assurant que chaque robot dispose des ressources nécessaires pour accomplir ses tâches sans interférence ni conflit avec d'autres robots. Pour la fiabilité et la sécurité, en fournissant un environnement de développement et d'exécution rigoureux, les RTOS contribuent à la fiabilité et à la sécurité des systèmes multirobots, réduisant les risques de pannes et de comportements imprévus.

Dans des applications telles que les drones collaboratifs, les robots de logistique en entrepôt, ou les équipes de robots de secours, l'utilisation d'un RTOS est cruciale. Par exemple, dans un entrepôt automatisé, les robots doivent se déplacer de manière coordonnée pour éviter les collisions tout en optimisant le chemin pour collecter et livrer des articles. Un RTOS permet de gérer ces tâches en temps réel, garantissant une exécution fluide et efficace des opérations.

Un RTOS joue un rôle essentiel dans l'optimisation de l'allocation des tâches dans un système MRTA. Il assure la gestion déterministe et prévisible des tâches, améliore la coordination entre les robots, et garantit une réponse rapide aux événements critiques, contribuant ainsi à la performance globale et à la fiabilité des systèmes multirobots.

IV.7. Conclusion

L'étude de l'allocation de tâches multirobots (MRTA) vise à optimiser l'allocation des tâches entre plusieurs robots pour améliorer l'efficacité opérationnelle. Pour résoudre ce problème complexe, les méthodes de recherche en MRTA explorent différents algorithmes. Les approches centralisées utilisent un seul contrôleur pour prendre des décisions d'allocation en analysant l'ensemble du système, tandis que les méthodes décentralisées permettent à chaque robot de

prendre des décisions localement en coopération avec d'autres. Les algorithmes de marché inspirés des enchères, où les robots « enchérissent » sur des tâches, sont également populaires. Des techniques de résolution de problèmes distribuées telles que les algorithmes de consensus peuvent également être appliquées. L'objectif de la recherche en MRTA est de trouver un équilibre entre la complexité des algorithmes, l'évolutivité et la qualité des solutions d'affectation pour optimiser les performances des systèmes multirobots.

Chapitre V
***Conception du VLiDAR3D et
d'AlMustaksheef3D***

Chapitre V : Conception d'AlMustaksheef3D et du VLiDAR3D

V.1. Introduction

Durant ce projet de développement de cluster de microrobot dédiés à l'intervention sur les sites caractérisés par un degré d'hostilité [306] vis-à-vis à une présence humaine, l'idée de base nécessite d'appréhender des méthodologies de conception multiples. Ce contexte est polyvalent, si l'on considère que son ingénierie fait invoquer toutes les compétences, du relatif au dessin du prototype du microrobot à son montage réel à la suite de l'impression 3D, sans oublier le montage d'environnement virtuel et logique du contexte [307] pour pouvoir lancer une quelconque simulation. A la fin de cette simulation, il est recommandé d'en tirer les conclusions.

V.2. Présentation du vLiDAR3D

Le module le plus recommandé pour ce genre de situation de reconnaissance des environnements de navigation, c'est le Lidar. Cet équipement a la possibilité de collecter un volume d'information aussi dense et de telle sorte que cette acquisition, puisse être remodeler afin de développer des modèles 3D du contexte physique où évolue le microrobot (Fig. V.1).



Fig. V.1 : Prototypes en voie de développement, AlMustaksheef3D, (a) Le microrobot. (b) le Lidar

Le modèle du Lidar proposé (Fig. V.2), se caractérise par une topologie avec tête active verticale. Cette orientation, lui procure une ouverture de balayage de pratiquement 180° sr (stéradian) avec deux degrés de liberté, une rotation dans le plan horizontal de plus de $\pm 180^\circ$ en mode continu et une rotation de plus de $\pm 120^\circ$ sur l'axe de la tête Emetteur/Récepteur laser.

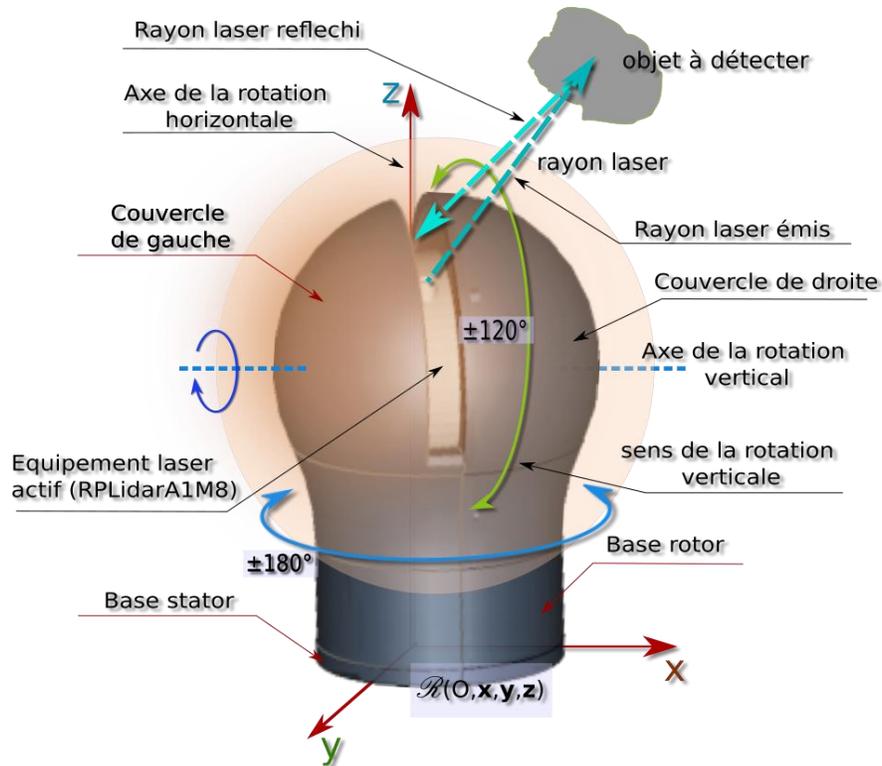


Fig. V.2 : Lidar développé (vLiDAR3D) et degrés de liberté

Le lidar (LiDAR) est un équipement utilisé pour interférer avec les objets exposés dans son environnement à l'aide d'un rayon laser émis à partir d'un émetteur jumelé avec une sonde réceptrice. L'aller-retour (émission-réception) du rayon laser par rapport à l'objet balayé, détermine le double de la distance séparant la sonde laser et un point spécifique (x, y, z) sur l'objet.

La collection de grand volume de points scrutés, représente un nuage de points (PC, Point Cloud). Ce PC est une forme de modèle de trois dimensions, sauf que le système sur lequel la sonde de balayage laser, doit être en mouvement autour de l'objet à examiner sinon que l'objet, tourne sur 360 degrés (balayage plan) autour de son axe, ceci pour permettre une première reconnaissance.

L'information déduite permet à toute machine sur laquelle est monté ce type de lidar, de l'utiliser pour plus de deux fins principales. La première, est l'acquisition de la forme scannée et la deuxième consiste en la détection d'obstacle et de sa forme. Un équipement LiDAR, procure à un robot le meilleur moyen pour se doter des deux fonctionnalités.

L'importance de cette fonction même, joue un rôle prépondérant dans les opérations SLAM (Simultaneous Localization And Mapping) où 'Cartographie et localisation simultanée' dans des missions de grandes importances, en grande partie dans des sites définies comme hostiles. Il est à

remarquer que ces sites dits difficiles sont généralement, les lieux touchés par des aléas majeures et où l'accès aux humains est tenté sous le risque imposant. L'intervention des machines robotisées évitent considérablement ces éventuelles pertes en vies humaines.

Le développement proposé, tente à optimiser les carences morphologiques du lidar, élément principal pour une tâche SLAM. La plupart des équipements équivalents fonctionnent en mode plan, alors que notre approche d'optimisation vise à augmenter cette dernière faculté en doublant l'angle de balayage. En d'autres termes, la modification se porte sur un angle sphérique de plus de 120° sr (Fig. V.2).

Certes, plus le champ d'appréhension est large, plus l'acquisition est valide et reflète les détails de l'objet scanné. Le traitement en arrière-plan sur le microcontrôleur est beaucoup lié au volume de données à traiter, aussi bien à leur compatibilité aux fins détails à scruter, la digitalisation 3D d'un lieu est sujette de cette qualité.

Pour la localisation d'un microrobot sur un lieu n'est pas totalement fonction des détails, alors la digitalisation 3D l'est. Ces deux opérations, sont exercées en même temps que le microrobot évolue dans son environnement. La détection d'obstacle par acquisition des limites d'objets environnants par rapport à un plan horizontal, jumelée avec une opération de digitalisation 3D sur 120° sr (Fig. V.3).

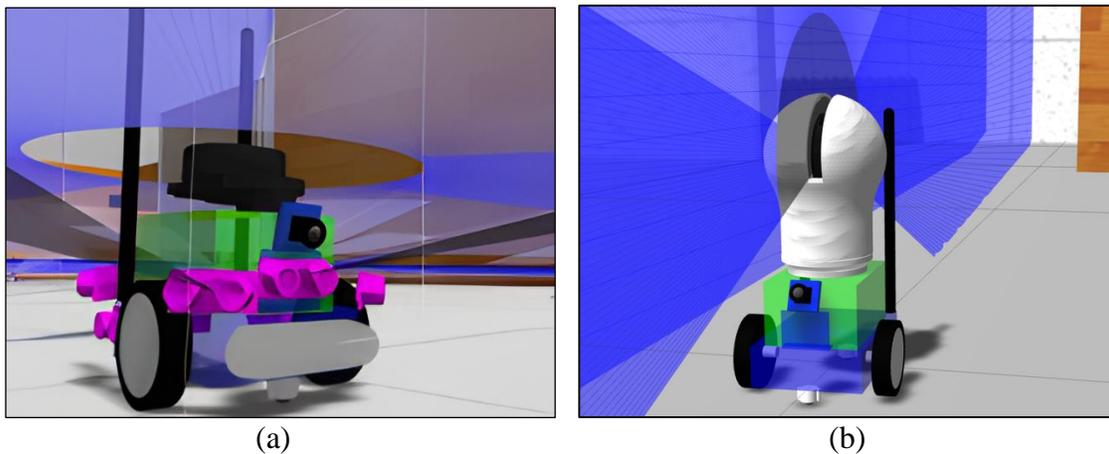


Fig. V.3 : Deux configuration de LiDAR horizontal (a) (plan sur 2π rd) et vertical (b) (sphérique sur 120° sr)

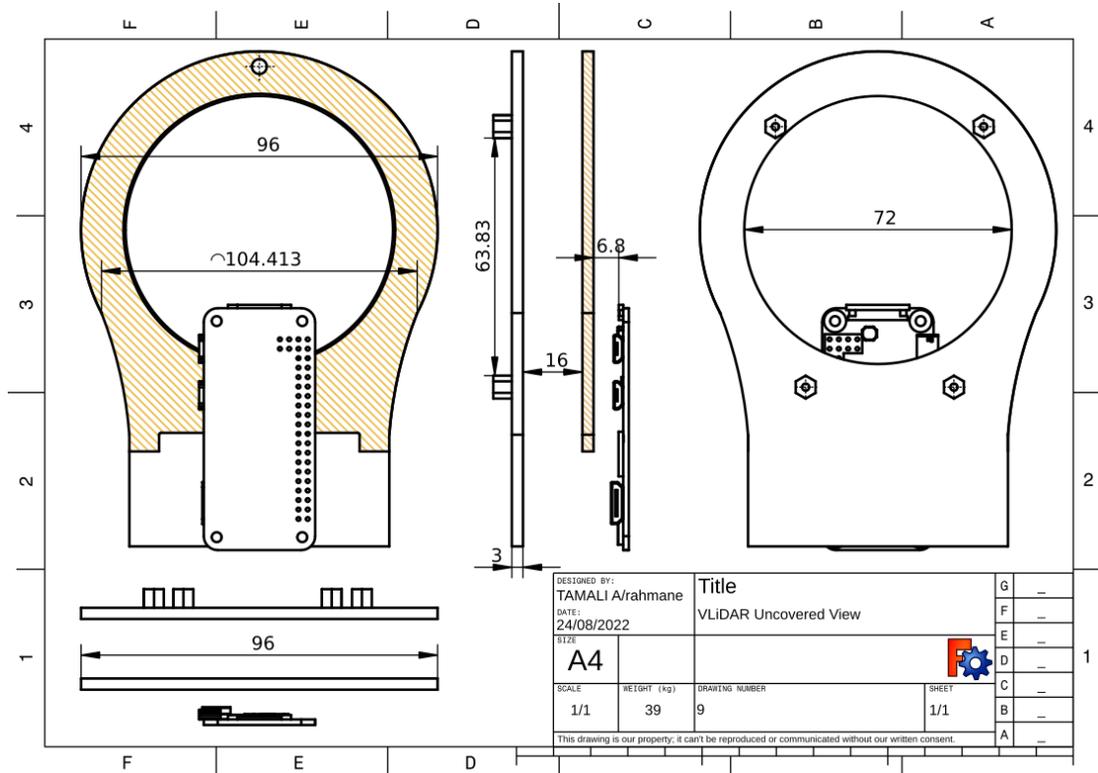


Fig. V.4 : Planche de dessin technique du vLiDAR3D proposé (Dessin exécuté sur FreeCAD¹)

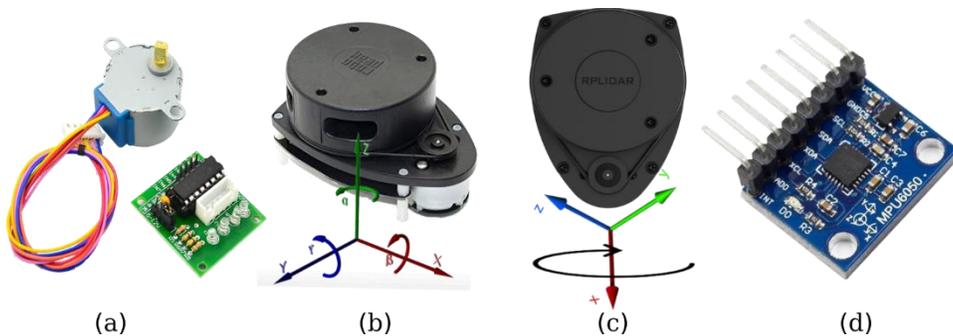


Fig. V.5 : Deux positions du RPLiDAR A1M8, (a) Moteur pas-à-pas, (b) LiDAR en horizontale (360° balayage plan), (c) Lidar en verticale (360 sr balayage sphérique), (d) Détecteur inertiel MPU6050

La figure V.4, représente des composants du vLiDAR3D proposé, support du microcontrôleur Raspberry Pi 2WH qui conduit toutes les opérations de détection à balayage laser et du détecteur inertiel IMU, Le vLiDAR3D comporte aussi un moteur pas-à-pas dans sa base. Le Lidar RPLiDAR A1M8 de SlamTec monté longitudinalement dans la partie tournante. Cette préférence dans le montage, suscite une correction dans les formules de calcul utilisées pour déterminer les distances captées, tel qu'illustré dans les figures V.5 (a), (b), (c) et (d).

¹ FreeCAD: Logiciel CAD Open-Source avec beaucoup de fonctionnalités

Le microcontrôleur, embarque un système d'exploitation Linux Ubuntu 20.04 LTS avec le nécessaire de composantes logiciels et bibliothèques nécessaires pour le contrôle et la commande des détecteurs et actionneurs montés sur le vLiDAR3D. Le code responsable pour la gestion totale est écrit en Python ou C/C++ et coexiste avec le middleware ROS (Robotics Operating System). La communication des données avec le microrobot sur lequel il est monté, est faite par les moyens offerts de manière câblée ou sans tel que le Wifi.

V.3. Modèle du microrobot AlMustaksheef3D

Le microrobot avec son système d'exploitation Ubuntu/ROS est programmé de telle sorte qu'il peut interchanger le contenu de l'information acquise durant une tâche de manière collaborative avec les autres agents similaires dans un même groupe dit cluster.

Le microcontrôleur qui traite les données dans chaque unité microrobot, est doté de processus ARM Cortex A72 Quad Cores, responsables de la gestion des fonctions suivantes :

- **Odométrie**, positionnement du véhicule dans son environnement. L'odométrie pour les microrobots implique l'utilisation de capteurs miniaturisés, tels que des encodeurs optiques et des IMUs (accéléromètres et gyroscopes), pour déterminer précisément leur position et orientation dans un environnement donné. À cette échelle, les microrobots doivent compenser les erreurs significatives dues au glissement des roues et aux interférences, souvent à l'aide de méthodes de fusion de capteurs comme le filtrage de Kalman. Les équations clés incluent la mise à jour de la position et l'orientation :

- $x_{\text{nouveau}} = x_{\text{ancien}} + d \cdot \cos(\theta)$

- $y_{\text{nouveau}} = y_{\text{ancien}} + d \cdot \sin(\theta)$

Où d est la distance parcourue calculée à partir du nombre de rotations des roues, et θ est l'orientation actuelle du robot. Ces techniques permettent aux microrobots de naviguer efficacement et d'exécuter des tâches précises même dans des environnements complexes ou restreints [2].

- **Navigation**, mouvement dans son environnement. La navigation des microrobots dans leur environnement repose sur des stratégies qui leur permettent de se déplacer de manière autonome et efficace. Cette capacité est essentielle pour effectuer des tâches dans des espaces restreints ou complexes où une intervention humaine serait difficile ou impossible. En utilisant des technologies telles que l'odométrie, la vision par ordinateur, et parfois la localisation et cartographie simultanées (SLAM), les microrobots peuvent détecter et éviter

les obstacles, suivre des itinéraires prédéfinis, et adapter leur trajectoire en temps réel en fonction des changements dans leur environnement. Ces capacités de navigation sont fondamentales pour des applications variées, allant de l'exploration de zones sinistrées à la maintenance de machines ou d'infrastructures, en passant par la surveillance environnementale et la livraison de micro-cargaisons [9].

- **Cartographie SLAM**, Collection de PC pour le montage de modèle 3D. L'idée de combiner la cartographie SLAM (Simultaneous Localization and Mapping) avec la collecte de données à partir de point-cloud (nuages de points) pour le montage de modèles 3D est très prometteuse, surtout dans les domaines de la robotique et de la modélisation environnementale. Le SLAM permet à un robot de naviguer dans un environnement inconnu tout en cartographiant cet espace en temps réel. En intégrant la collecte de données en point-cloud, qui capturent des milliers de points par seconde pour former une image 3D précise de l'environnement, on peut non seulement améliorer la précision de la carte générée par SLAM, mais aussi créer des modèles 3D détaillés pour diverses applications, telles que la réalité augmentée, la planification urbaine ou la conservation du patrimoine. Cette approche offre une richesse de détails et une précision spatiale qui sont cruciales pour les systèmes autonomes nécessitant une interaction avancée avec leur environnement [9].
- **Télécommunication**, Echange d'information avec l'opérateur humain ou entre les microrobot. L'intégration des télécommunications dans les systèmes de microrobots pour faciliter l'échange d'informations avec des opérateurs humains ou entre les microrobots eux-mêmes est une avancée cruciale pour l'automatisation et la robotique collaborative. Cette capacité de communication permet aux microrobots de travailler de manière plus efficace et autonome, en transmettant des données en temps réel sur leur environnement ou leur état, et en recevant des instructions ou des mises à jour de mission à distance. Pour les opérateurs humains, cela signifie une meilleure capacité à surveiller, contrôler et intervenir si nécessaire, améliorant ainsi la sécurité et l'efficacité des opérations. La communication inter-robots permet également à ces systèmes de collaborer sur des tâches complexes, en partageant des informations sensorielles et des données de localisation, ce qui est essentiel pour des tâches comme l'exploration de zones inaccessibles ou la réalisation de missions de surveillance coordonnées [19].

- **Sécurité du lien**, avec le cluster. L'assurance de la sécurité des communications dans un cluster de microrobots est essentielle pour maintenir l'intégrité et la fiabilité des opérations robotiques collaboratives. Cette sécurité renforcée est cruciale pour prévenir les interférences malveillantes ou accidentelles, qui pourraient compromettre les missions en altérant les données échangées entre les robots ou avec la station de contrôle. En intégrant des protocoles de cryptage robustes et des méthodes d'authentification sécurisées, on peut protéger efficacement le flux d'informations, garantissant que seules les entités autorisées puissent accéder et manipuler les données de mission. Cette mesure est particulièrement importante dans des applications sensibles comme la surveillance de zones sécurisées, les interventions en environnements hostiles ou les opérations où les microrobots doivent fonctionner de manière autonome sur de longues périodes sans supervision directe [19].

Pour des raisons de simulation sous environnement tels que RVIZ, GAZEBO, OnShape ou Copellia-Sim, la recommandation nécessite l'intégration du modèle 3D dans la scène des agents montés, chacun sur un support mobile (un agent microrobot du cluster). De manière général (Fig. V.6), un script URDF², [308], SDL. Sa position est définie par rapport à un référentiel lié à la base (Fig. V.7), dit `base_link` sur laquelle, il est monté.

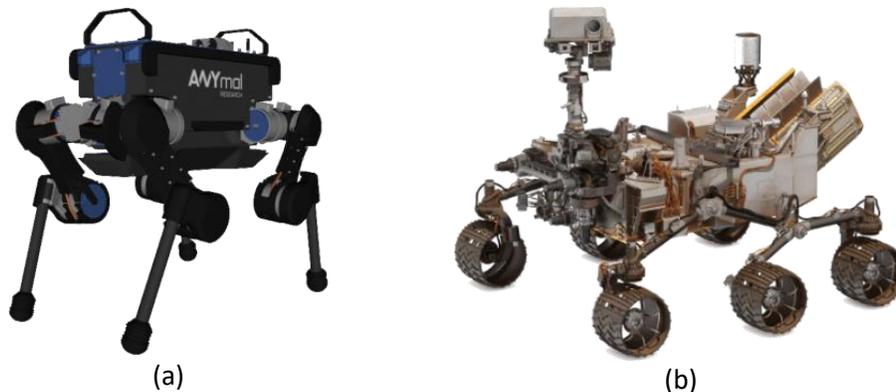


Fig. V.6 : Représentation URDF 3D du robot (a) ANYMal B d'ANYBOTICS, (b) Véhicule ROVER M2020, Crédit NASA/JPL-Caltech. Véhicules conçus pour des missions sur sites hostiles

² URDF : Unified Robot Description Format

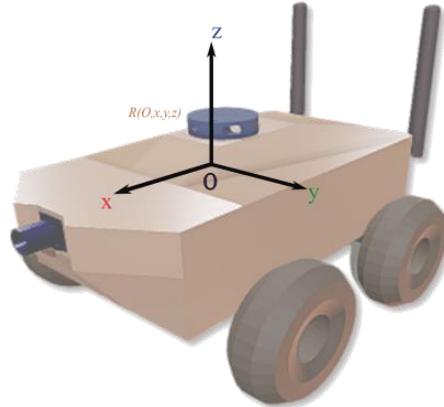


Fig. V.7 : Référentiel lié à la base du microrobot

URDF est un langage script XML, utilisé pour décrire une scène de simulation, en utilisant des balises, permettant la description exacte et détaillée du robot en action, de son environnement et de la topologie du lieu sur lequel évolue le robot. URDF est un langage script XML, utilisé pour décrire une scène de simulation, en utilisant des balises, permettant la description exacte et détaillée du robot en action, de son environnement et de la topologie du lieu sur lequel évolue le robot. VLiDAR3D (Fig. V.8) est décrit sous écriture URDF et est monté à un microrobot spécifique, NVidia NANOBOT (Fig.V.3b). D'autres langages XML, encore plus développés, peuvent être utilisés pour décrire en détail les composantes d'une scène de simulation.

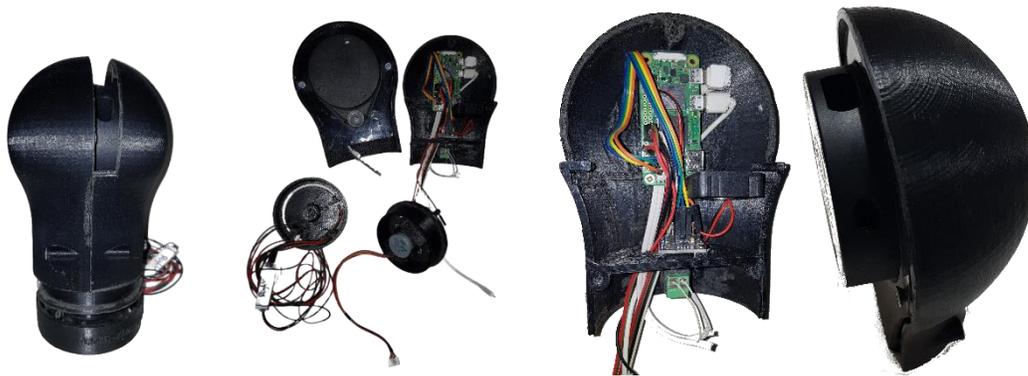


Fig. V.8 : vLiDAR3D conçu, imprimé et développé avec ses composantes

Pour des raisons de manipulation physique du microrobot, soit en simulation sous un environnement comme Gazebo ou en réalité, les composantes doivent avoir chacune un rattachement spatial par rapport au référentiel (absolu) de base $R(O,x,y,z)$ du nœud `base_link`. A cet effet, tout le corps du microrobot, peut être assimilé à une arborescence topologique de joint défini dans l'espace par un nœud (Fig. V.9). A ce dernier est affecté un référentiel relatif d'où,

tout mouvement peut être traité en dépendance au mouvement global (lié au `base_link`) du microrobot.

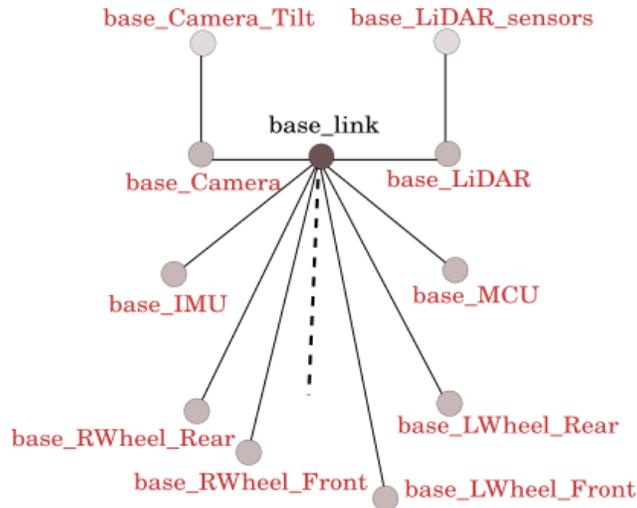


Fig. V.9 : Topologie du microrobot à décrire en modèle XML-URDF

Des logiciels tels que Gazebo, RViz sont les boîtes à outils à large usage pour la validation des approches. Les morphologies des microrobots proposés, répondent aux exigences des missions à entreprendre. Les conditions de confiance, de poids, d'autonomie, d'utilité, d'interopérabilité et de maintien du lien de connexion réseau, sont majeures et obligatoires.

Les plateformes en question sont des environnements logiciels complets et encore plus Open-Source, elles ne nécessitent qu'une bonne adhérence avec les milliers de bibliothèques à installer et configurer selon les modes '**Simulation**' ou '**Réel**'.

La configuration microrobot adoptée 'AlMustaksheef3D' est une variante adaptée du NVIDIA Nanobot surélevée par le vLiDAR3D et est équipée de caméra PiCamera (ou PIXI Cam intégrant de l'IA et programmable par Micro-Python). Au total, d'autres équipements s'ajouteront en termes de charge utile. Cette configuration est le fruit d'un développement étalé sur beaucoup d'années, durant lesquelles, la forme a été remodelée afin de répondre aux critères de l'équilibre mécanique et de l'espace réservé au port de batterie pour la question autonomie (Fig. V.10).



Fig.V.10 : Versions du microrobot AlMustaksheef3D durant son développement

Quelques planches de dessin technique sont exposées en guise de validation, la version finale du casing du vLiDAR3D tel qu'il a été préparé pour des raison d'impression 3D est donnée par les figures V.11, V.12, V.13, V.14 et V.15 suivantes.

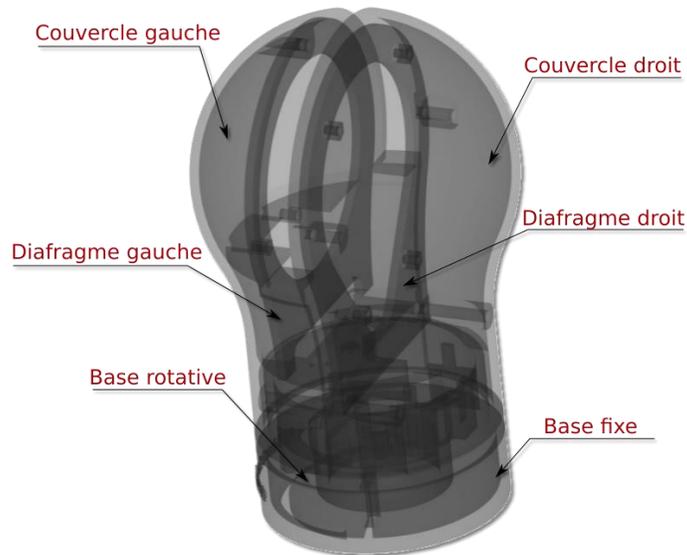


Fig. V.11 : Morphologie et composition générale du vLiDAR3D sans équipements

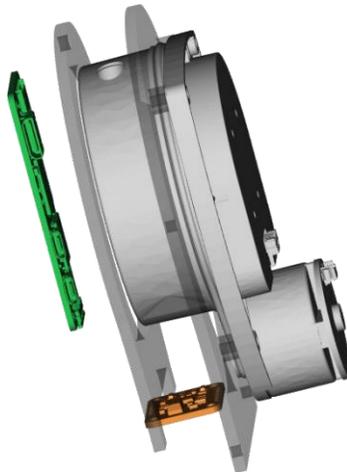


Fig. V.12 : Vue détail, principaux composants actifs

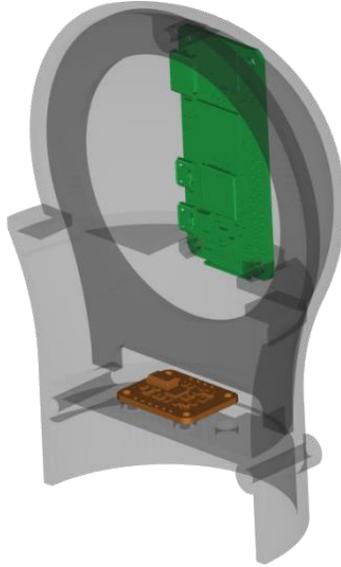


Fig. V.13 : Vue détail, MCU et IMU

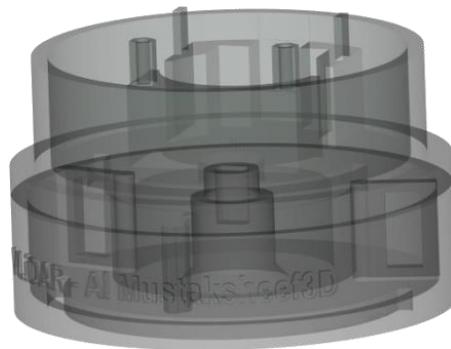


Fig. V.14 : Vue détail, base fixe

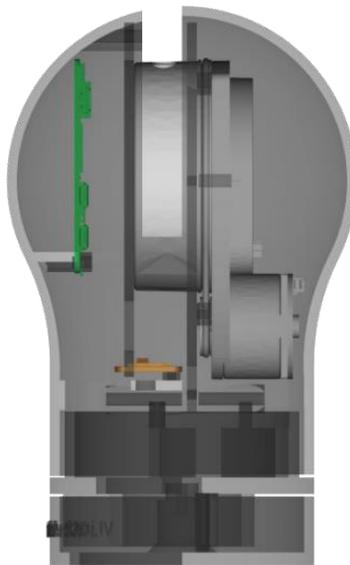


Fig. V.15 : Vue de côté, détail de l'appareil

En somme, nous tentons à réduire au maximum de prix total de la réalisation du microrobot et de ses modules principaux. Des logiciels Open-Source (Figure V.16), sont utilisés à cet effet pour le développement CAD (OnShape, FreeCAD), l'impression 3D (CURA), le développement des PCB (KiCAD) et l'environnement de programmation des codes (Microsoft Code) ainsi que ceux utilisés pour la simulation (GASEBO, RViz, RQT, Nvidia Isaac Sim, Webot, Coppelia Sim).

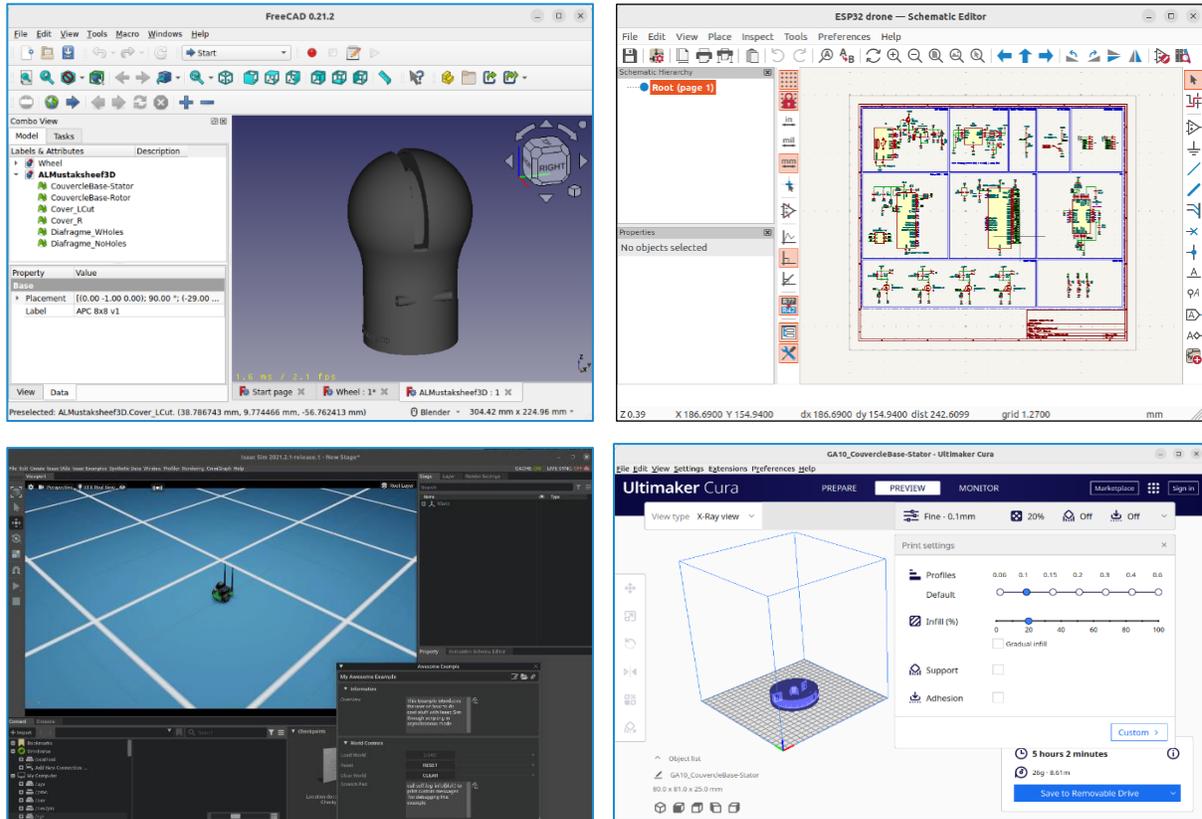


Fig. V.16 : Logiciels GPL Utilisés pour le développement

V.4. Conclusion

En conclusion, la conception et la modélisation de microrobots pour une utilisation dans des environnements hostiles représentent une avancée cruciale dans les domaines des secours en cas de catastrophe et des interventions humanitaires. Ces technologies mettent en lumière la nécessité de pousser encore plus loin les frontières de l'innovation pour surmonter les défis posés par des environnements dangereux et imprévisibles. En offrant une alternative agile et moins risquée aux interventions humaines, les microrobots tirent parti des dernières avancées en robotique, communication, Internet des Objets (IdO), et sécurité des réseaux pour proposer des solutions

efficaces et adaptatives. De plus, l'étude approfondie des comportements collaboratifs et des stratégies de navigation améliore significativement l'efficacité opérationnelle de ces dispositifs. Les recherches actuelles et les cas d'utilisation démontrent clairement le potentiel des clusters de microrobots pour améliorer la gestion des catastrophes et sauver des vies.

Pour aller de l'avant, il sera essentiel d'accentuer la recherche sur l'intégration des systèmes d'intelligence artificielle pour améliorer l'autonomie des microrobots, permettant une prise de décision rapide et efficace en situation de crise. De plus, le développement de protocoles de communication sécurisés et résistants aux interférences est crucial pour garantir le succès des missions. Enfin, une attention particulière devra être portée à la conception de systèmes collaboratifs entre microrobots pour optimiser les opérations de recherche et de sauvetage, maximisant ainsi l'efficacité des interventions et la sécurité des zones touchées. Ces avancées pourraient transformer radicalement la manière dont nous répondons aux catastrophes naturelles et sauver davantage de vies humaines à l'avenir.

Ainsi, L'étude des approches algorithmiques, des technologies de transmission et des méthodologies de simulation a fourni des informations précieuses pour la conception et le déploiement de microrobots dans des environnements hostiles. Les résultats de la recherche soulignent l'importance des clusters de microrobots orientées vers l'IdO pour améliorer la sécurité, la reconnaissance des états du système et la collaboration entre les entités robotiques.

Chapitre VI
Etude de cas

Chapitre VI : Etude de cas

VI.1. Introduction

Cette étude se consacre à améliorer les capacités de recherche et de sauvetage grâce à la simulation de la collaboration entre robots dans des environnements hostiles, en utilisant le Framework ROS. L'objectif est d'améliorer les capacités collectives d'une équipe de robots, leur permettant de travailler en tandem de manière transparente pour rechercher et localiser rapidement les victimes et en digitalisant en 3D le milieu étudié. Notre modèle évalue l'efficacité de la collaboration multirobots en comparant, en l'absence de travail d'équipe, le travail apparaîtra chaotique et désorganisé par rapport à ce que l'on verra dans le cas du travail d'équipe dans un environnement considéré hostile, en utilisant des outils, des algorithmes et des principes d'optimisation avancés. Les résultats de cette étude pourraient influencer la conception et la mise en œuvre de futurs systèmes robotiques et cobotiques pour les opérations de recherche et de sauvetage. En enquêtant et en analysant rigoureusement, nous apportons des informations précieuses au domaine plus large de la robotique et de la réponse aux catastrophes. Les sections suivantes approfondiront les fondements théoriques, la méthodologie, les résultats et les implications de notre étude de simulation, offrant une compréhension globale des avantages et des défis potentiels liés à l'utilisation de la collaboration multirobots dans la recherche hostile et les opérations de sauvetage.

VI.2. Le Framework ROS

ROS, ou Robot Operating System, est un Framework middleware open source conçu pour faciliter le développement de logiciels pour les applications robotiques. Malgré son nom, ROS n'est pas un système d'exploitation au sens traditionnel du terme et on peut le considérer comme étant un système d'exploitation pour les robots ; au lieu de cela, il fournit des services et des outils pour aider à gérer l'abstraction matérielle, les pilotes de périphériques, la communication entre les processus, etc. Il sert de plate-forme flexible et modulaire, offrant aux roboticiens un moyen standardisé de créer et de partager du code pour une large gamme de systèmes robotiques. Il agit comme un middleware, fournissant une couche de communication entre les différents composants d'un système robotique. Cela facilite l'échange d'informations entre les capteurs, les actionneurs et divers modules logiciels [309].

ROS extrait les détails du matériel, permettant aux développeurs d'écrire du code qui peut être facilement porté sur différentes plates-formes robotiques. Cette abstraction simplifie le processus de développement et favorise la réutilisation du code dans divers systèmes robotiques.

Comme gestion des packages, ROS organise le code en packages, chacun encapsulant des fonctionnalités spécifiques. Ces packages peuvent être facilement partagés et réutilisés par la communauté robotique, favorisant ainsi la collaboration et accélérant le développement (Fig .VI.1).

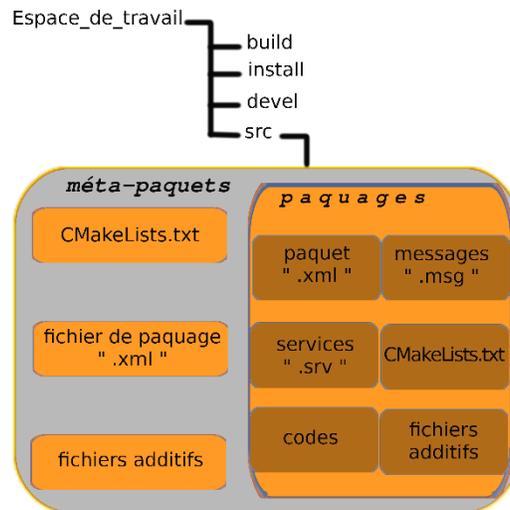


Fig.VI.1 : Le système de fichier de ROS

L'infrastructure de communication de ROS est sous le principe d'une architecture de publication-abonnement ou serveur-client, permettant la communication entre différents nœuds (modules logiciels) dans un système distribué. Les nœuds peuvent publier des données (par exemple, des relevés de capteurs) sur des sujets (topics) spécifiques, et d'autres nœuds peuvent s'abonner à ces sujets pour recevoir les données. Le middleware ROS dispose d'une communauté vaste et active de développeurs, de chercheurs et de roboticiens. Cette communauté contribue à un vaste référentiel de packages, de bibliothèques et de documentation, permettant aux utilisateurs de trouver plus facilement des ressources et des solutions aux problèmes courants.

Il est livré avec un ensemble complet d'outils pour le débogage, la visualisation, la simulation et l'analyse. Les exemples incluent Rviz pour la visualisation, Gazebo pour la simulation et divers outils de ligne de commande pour gérer les composants ROS.

Ce dernier, il est compatible avec divers systèmes d'exploitation, notamment Linux (Ubuntu étant la distribution la plus couramment utilisée), macOS et, dans une certaine mesure, Windows.

Cette flexibilité permet le déploiement de ROS sur une large gamme de plates-formes informatiques.

VI.2.1. Utilisation de ROS pour les systèmes multirobots

ROS est particulièrement bien adapté aux systèmes multirobots en raison de son architecture modulaire et de ses capacités de communication. Voici comment ROS facilite le développement et l'exploitation de systèmes multirobots [309] :

- Architecture de nœud : dans ROS, chaque composant robotique (capteur, actionneur, contrôleur, etc.) est généralement représenté sous forme de nœud. Les nœuds communiquent en publiant et en s'abonnant à des sujets. Dans un système multirobot, chaque robot peut être considéré comme un ensemble de nœuds, permettant un traitement distribué et parallèle.
- Paradigme de communication : le modèle de communication publication-abonnement de ROS est très propice aux systèmes multirobots. Les nœuds peuvent communiquer entre eux de manière asynchrone, permettant aux robots de partager des informations, de coordonner leurs actions et de collaborer efficacement.
- Architectures centralisées ou décentralisées : ROS prend en charge les architectures centralisées et décentralisées pour les systèmes multirobots. Dans une approche centralisée, un nœud central peut coordonner les actions de tous les robots. Dans une approche décentralisée, les robots communiquent directement entre eux et prennent des décisions collectivement.
- Modularité des packages : les packages ROS encapsulent des fonctionnalités spécifiques et ces packages peuvent être partagés entre différents robots. Cela favorise la réutilisation du code et accélère le développement de systèmes multirobots.
- Outils de simulation : ROS s'intègre à des outils de simulation comme Gazebo, permettant aux développeurs de simuler le comportement de plusieurs robots dans un environnement contrôlé. Ceci est inestimable pour tester et affiner les algorithmes avant de les déployer sur des robots physiques.
- Prise en charge des systèmes distribués : ROS fournit des mécanismes pour exécuter des nœuds sur différentes machines, facilitant ainsi le déploiement de systèmes multirobots dans des environnements informatiques distribués.

- Transmission de messages : ROS utilise un mécanisme de transmission de messages pour la communication. Les messages peuvent être personnalisés pour inclure des informations telles que des données de capteur, des commandes de contrôle ou des mises à jour de localisation. Cette flexibilité est essentielle pour transmettre les diverses informations nécessaires dans les scénarios de collaboration multirobots.

En résumé, ROS sert de plate-forme puissante et polyvalente pour le développement de systèmes multirobots, offrant les outils et l'infrastructure nécessaires à une communication, une coordination et une collaboration efficaces entre les entités robotiques.

VI.3. Préparation de la scène

Divers outils, notamment OnShape Assembly propriétaire et le module complémentaire Phobos open source dans Blender, FreeCAD de dessin avec expertise des objets 3D, sont utilisés pour les tâches préparatoires avant la simulation sous ROS. Les scripts XACRO, SDF ou URDF sont essentiels pour développer des modèles de robots 3D avec articulations pour la simulation Gazebo/RVIZ. Le prototype JETBOT de NVIDIA sert de micro-robot modélisé avec un URDF (SDF) mis à jour pour répondre aux besoins de l'étude. Les éléments à cette étape sont cruciaux pour mettre en place une scène de simulation compatible et conforme à notre approche :

- L'espace réel est défini par un emplacement spécifique (les dimensions d'un appartement avec des limitations spatiales) en 3D conçu sur Gazebo pour obtenir un fichier ".world" d'extension (voire les figures VI.2 et VI.3) [310-312].
- Le groupe de MR proposé est une formation de trois agents similaires [313].
- Possibilité du cas hétérogène [314-316].
- Cible, un humain blessé comme représente la figure VI.4 (modèle 3D d'un humain, complexité du comportement de la cible).

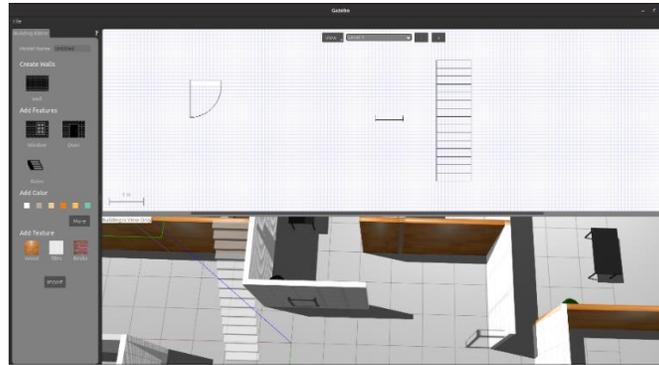


Fig.VI.2 : Outil éditeur de bâtiments de Gazebo

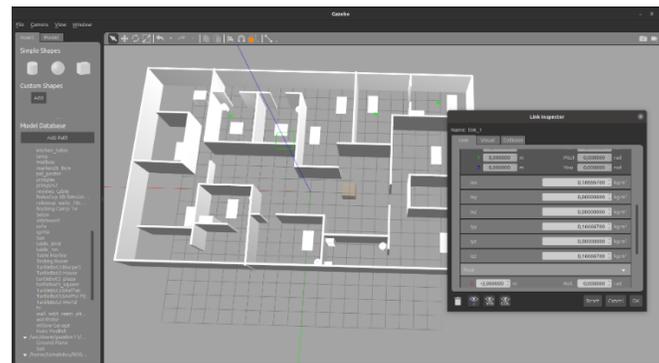


Fig.VI.3 : Outil éditeur de modèles de Gazebo

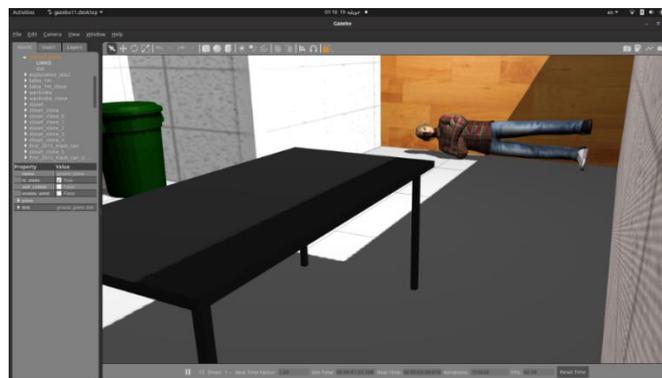


Fig.VI.4 : Modèle 3D d'un humain pour émuler un humain blessé

Montage standard :

- Économie et autonomie énergétique pour une charge utile donnée (implémentation du ROS Battery Plug-In (Figure VI.5) au sein des joints URDF avec sa bibliothèque compilée compatible).
- Assurer la stabilité des liens de communication à l'aide de la technique GMLA et du plug-in de connectivité ROS Radio avec le cluster.

- Améliorez les capacités de collaboration pour améliorer les performances en implémentant les plug-ins, les règles et les algorithmes nécessaires, comme illustré dans l'algorithme 1.
- IA basée sur les améliorations d'algorithmes (code AI Logic intégré).
- Taille et volume minimum de l'agent (à des fins d'extension d'essaim).

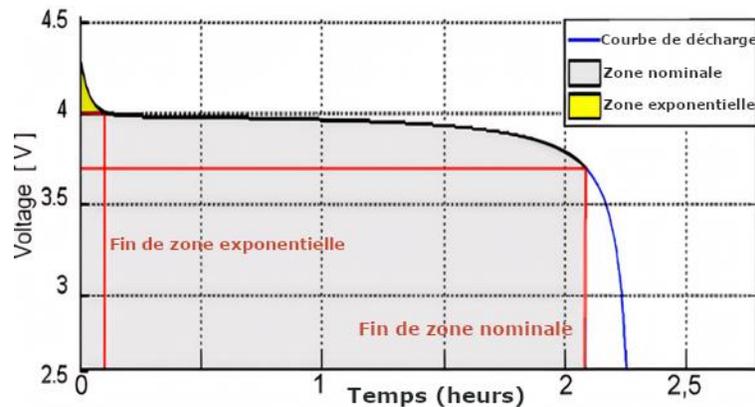


Fig.VI.5 : Caractéristique de décharge de courant nominal (0,65 A)

Exigences de la scène :

- Le cluster intègre trois microrobots (Agents) notés, chacun équipé de roues motorisées pour la locomotion (Différentiel, Conducteur).
- Chaque unité d'agent est dotée de capacités de transmission pour la transmission de données et le maintien des liens de communication.
- L'autonomie de chaque agent est soutenue par une batterie embarquée, fournissant une puissance suffisante pour ses activités au sein du cluster.
- Chaque agent possède une base algorithmique d'IA intégrée, lui permettant de prendre des décisions à des fins collaboratives et individuelles.
- Chaque unité d'agent du cluster est équipée d'équipements essentiels, notamment une caméra RGBD, un Lidar, un pilote de moteur, des actionneurs et des capteurs (Figure VI.6).
- Le groupe de microrobots fonctionne selon deux modes : le mode supervisé et le mode Collaboration.

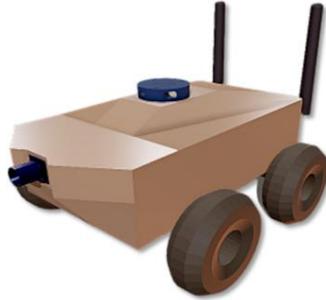


Fig.VI.6 : AL Mustaksheef3D, développement d'un robot à roues

Notre objectif est de localiser et d'aider efficacement les victimes dans des situations difficiles grâce à la MRTA collaborative. Notre approche implique la mise en œuvre de la numérisation 3D à l'aide du Lidar ou d'une caméra de profondeur, garantissant des résultats optimaux à des fins de recherche et de SLAM. L'environnement ROS sert à gérer le contexte et, en plus de l'interopérabilité, ROS peut interagir avec diverses plates-formes dédiées à la simulation des problèmes MRTA [79, 307, 317-319]. Le module intégré ROS.MSG facilite l'échange de messages entre microrobots, la conversion des données IMU en ODOM permettant la validation et le vote des mouvements d'agents dans les zones encore inoccupées (Figure VI.7).

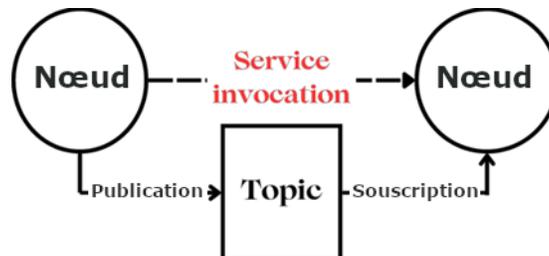


Fig.VI.7 : Bases et concepts de ROS

VI.4. Processus de collaboration et d'affectation de tâches utilisé

Réaliser un travail d'équipe coordonné entre les robots avec une répartition des tâches en temps réel nécessite un cadre décentralisé [320] qui couvre la conscience résiliente des robots, le contrôle des mouvements de bas niveau et la planification des tâches de haut niveau [321, 322]. Une gestion efficace de la localisation dans les réseaux multirobots est essentielle à la collaboration, où la prise de décision et la communication jouent un rôle central dans la répartition des missions, présentant des défis substantiels dans l'échange de données entre les robots et le poste d'exploitation [323].

Dans les approches autoorganisées et décentralisées, les nœuds robots individuels prennent des décisions de manière indépendante avec un respect limité pour les autres agents [324]. Les

décisions dans ces approches sont fréquemment influencées par des phénomènes naturels ou du monde réel, comme le soulignent [325], notamment l'essaim d'abeilles, la stratégie de marché, l'intelligence des essaims et la colonie de fourmis. Ces inspirations donnent naissance à des comportements collectifs complexes nés d'interactions locales entre de nombreux agents aux comportements simples. Dans de telles méthodologies, les capteurs jouent un rôle central en collectant activement les connaissances locales pour les partager au sein du cluster, comme le soulignent [216, 326]. L'utilisation collaborative de capteurs facilite l'accumulation des connaissances nécessaires relatives à un objectif primordial de ces procédures (Figure VI.8).

Les robots doivent être capables de comprendre les tâches qu'ils doivent effectuer en collectant des données via des capteurs et en utilisant un code spécifique pour prendre des décisions. Le haut débit, la portée, la puissance et les débits de données sont essentiels à la performance du système. La communication point à point est la forme la plus élémentaire et le choix du support de transmission dépend du type d'informations échangées.

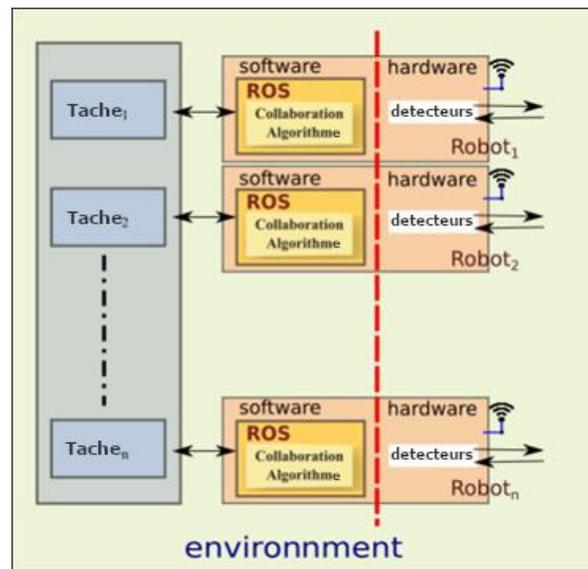


Fig.VI.8 : Architecture du cluster

VI.5. Énoncé du problème

L'évolution du robot sur des sites hostiles peut, en quelque sorte, être supposée être une progression du véhicule formant un chemin sur une surface contournée par un ensemble de N points dans un espace défini dans un plan délimité par un polygone fermé, où $P = P_1, P_2, \dots, P_N$ définir un Poly-point ou un ensemble de N points (Figure VI.9) :

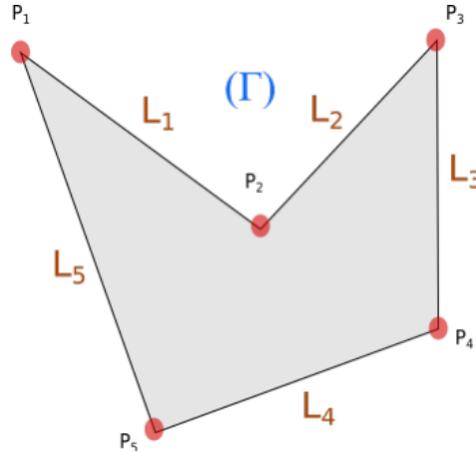


Fig.VI.9 : N-gon d'une zone numérisée

Normalement chaque point est localisé sur le plan (Γ) par ses coordonnées cartésiennes x_i et y_i . Une équation unifilaire peut s'écrire sous la forme :

$$y = a \cdot x + b \tag{Equ.VI.1}$$

a et b sont deux paramètres liés à la ligne $i^{ème}$ (pente et ordonnée à l'origine) dans la polyligne définie par les composants P. En utilisant les coordonnées de chaque couple de points, les paramètres a et b associés sont obtenus selon (en utilisant la règle de Cramer) :

$$a = \frac{y_1 - y_2}{x_1 - x_2} \tag{Equ.VI.2}$$

$$b = \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \tag{Equ.VI.3}$$

Ainsi, (Equ.VI.1) peut être comme suit pour une équation équivalente en ligne (L_1) entre deux points $P_1(x_1, y_1)$ et $P_2(x_2, y_2)$.

y est donné par :

$$y = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} \cdot x + \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} & \begin{cases} x_1 \leq x < x_2, \forall x_1 - x_2 < 0 \\ x_2 \leq x < x_1, \forall x_1 - x_2 > 0 \\ y_1 \leq y < y_2, \forall y_1 - y_2 < 0 \\ y_2 \leq y < y_1, \forall y_1 - y_2 > 0 \end{cases} \\ 0 & \text{ailleurs} \end{cases} \tag{Equ.VI.4}$$

Maintenant, nous devons construire une équation polygonale en utilisant une combinaison d'équations à plusieurs lignes.

Nous avons N points, ce qui implique que le nombre d'équations droites est $N(L_1, L_2, \dots, L_N)$.

La formule du n-gon (polygone) est donnée par :

$$Y = Y_1 + Y_2 + \dots + Y_N = \sum_{i=1}^{N-1} Y_i + Y_N \tag{Equ.VI.5}$$

Qui est Y_1, Y_2, \dots, Y_N ou simplement Y_i, Y_N où $i \in \mathbb{N}$ nombre naturel strictement positif, l'ensemble Y_i avec $i \in \mathbb{N}$ représente l'équation de chaque ligne :

$$Y_i = y = \begin{cases} \frac{y_i - y_{i+1}}{x_i - x_{i+1}} \cdot x + \frac{x_i y_{i+1} - x_{i+1} y_i}{x_i - x_{i+1}} & \begin{array}{l} x_i \leq x < x_{i+1}, \forall x_i - x_{i+1} < 0 \\ x_{i+1} \leq x < x_i, \forall x_i - x_{i+1} > 0 \\ y_i \leq y < y_{i+1}, \forall y_i - y_{i+1} < 0 \\ y_{i+1} \leq y < y_i, \forall y_i - y_{i+1} > 0 \end{array} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{Equ. VI.6})$$

Equation valable pour $1 \geq i \geq N - 1$, et la dernière ligne (L_N) :

$$Y_N = y = \begin{cases} \frac{y_1 - y_N}{x_1 - x_N} \cdot x + \frac{x_1 y_N - x_N y_1}{x_1 - x_N} & \begin{array}{l} x_1 \leq x < x_N, \forall x_1 - x_N < 0 \\ x_N \leq x < x_1, \forall x_1 - x_N > 0 \\ y_1 \leq y < y_N, \forall y_1 - y_N < 0 \\ y_N \leq y < y_1, \forall y_1 - y_N > 0 \end{array} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{Equ. VI.7})$$

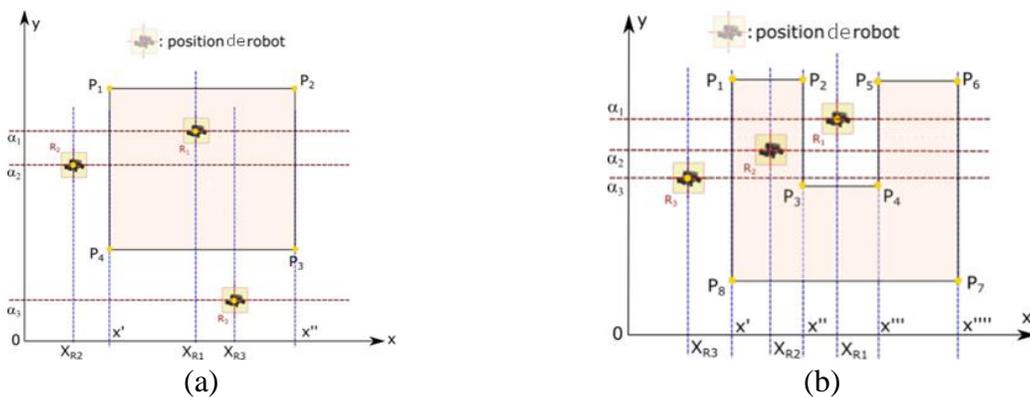
Pour $x, y, \alpha \in \mathbb{R}$ et $f(x) = y = \alpha$. La zone à l'intérieur du polygone irrégulier peut être définie comme le résultat de :

$$\text{Area}_Y = \int_{\mathbb{R}} \alpha |_{f(x)=\alpha=Y=\{x';x'';\dots;x^n;x^{n+1}\}} d\alpha \quad (\text{Equ. VI.8})$$

avec $x^n, x^{n+1} \in \gamma(x)$

Considérons des expressions telles que $n = 2m + 1 |_{m \in \mathbb{N}}$, où $x^n < x^{n+1}$ et $\gamma(x)$ représentant le domaine de variation de la variable x . Par conséquent, la zone explorée par le $k^{\text{ème}}$ robot peut être désignée par $Area(k)$, où $k=1, 2, \dots, R$ et R est le nombre total d'agents de cluster.

Nous introduisons la relation $g_k(\alpha, x)$, qui satisfait la condition $f(x) = \alpha = Y = \{x'; x''; x'''; \dots; x^n; x^{n+1}\}$. Cette relation, notée g_k , aide à déterminer si le robot k est situé ou non dans la zone désignée $Area(k)$. Pour évaluer si un robot se trouve à l'extérieur ou à l'intérieur d'une zone, nous pouvons analyser quatre situations en forme de zone illustrant les cas les plus probables, comme le montrent les figures VI.10a, VI.10b, VI.10c et VI.10d.



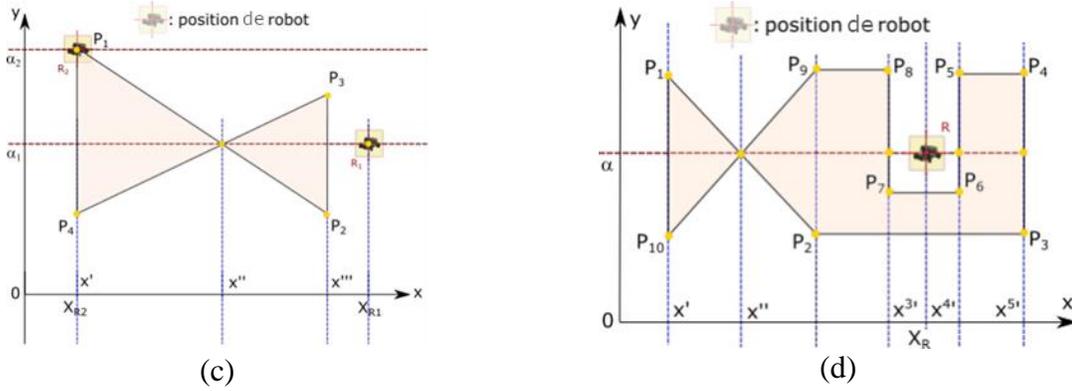


Fig.VI.10 : Scénarios potentiels pour la posture du robot, (a) Cas simple avec 3 robots, (b) Cas complexe avec 3 robots, (c) Cas singulier, (d) Cas insoluble

Dans le premier scénario (Figure VI.10a), le $robot_1$ est situé dans la zone Z , alors que les autres robots ($robot_2$ et $robot_3$) ne le sont pas. La formulation de cette situation implique des expressions telles que $g_1(\alpha_1, x) = g_2(\alpha_2, x) = Z\{x', x''\}$ et $g_3(\alpha, x) = \emptyset$.

Ainsi, en comparant avec x' et x'' , nous pouvons tirer les conclusions suivantes :

- Si $g_k(\alpha, x) = \emptyset \Rightarrow$ le $robot_k$ est hors de la zone Z .
- Si $x' \leq x_{R_k} \leq x''$ et $g_k(\alpha, x) \neq \emptyset \Rightarrow$ le $robot_k$ est dans la zone Z .
- Si $x_{R_k} < x'$ ou $x'' < x_{R_k}$ et $g_k(\alpha, x) \neq \emptyset \Rightarrow$ le robot est hors zone Z .

Dans le deuxième scénario (Fig.VI.10b), nous avons $g_1(\alpha_1, x) = g_2(\alpha_2, x) = g_3(\alpha_3, x) = Z\{x', x'', x''', x''''\}$. $robot_2$ se trouve dans la zone où $x' \leq x_{R_2} \leq x''$, alors que les autres ne le sont pas. Plus précisément, $x'' < x_{R_2} < x'''$ et $x_{R_2} < x'$, conduisant aux déterminations suivantes :

- Si $x^n \leq x_{R_k} \leq x^{n+1} \Big|_{n=2m+1, \forall m \in \mathbb{N}}$ et $g_k(\alpha, x) \neq \emptyset \Rightarrow$ le $robot_k$ est dans la zone.
- Si $x^n < x_{R_k} < x^{n+1} \Big|_{n=2m+2, \forall m \in \mathbb{N}}$ et $g_k(\alpha, x) \neq \emptyset \Rightarrow$ le $robot_k$ est hors de la zone.

La troisième situation (Figure VI.10c) englobe deux cas singuliers, où $g_1(\alpha_1, x) = \{x', x'', x'''\}$ et $g_2(\alpha_2, x) = \{x'\}$. La présence du robot dans la zone ne peut être déterminée que dans les cas où $|g_k|=3$ pour le $robot_1$ et $|g_k|=1$ pour le $robot_2$:

- Si $x' \leq x_{R_k} \leq x'''$ et $g_k(\alpha, x) \neq \emptyset$ où $|g_k|=3 \Rightarrow$ le $robot_k$ est dans la zone.
- Si $x' = x_{R_k}$ et $g_k(\alpha, x) \neq \emptyset$ où $|g_k|=1 \Rightarrow$ le $robot_k$ est dans la zone.

Dans le dernier cas (Figure VI.10d), il est impossible de savoir si le robot est dans la zone ou non en utilisant la relation g_k , où $g_k(\alpha, x) \neq \emptyset$, et $|g_k|=n \Big|_{n=2m+5, \forall m \in \mathbb{N}}$.

VI.6. Algorithme d'optimisation

L'organigramme suivant illustre le processus global (Figure VI.11).

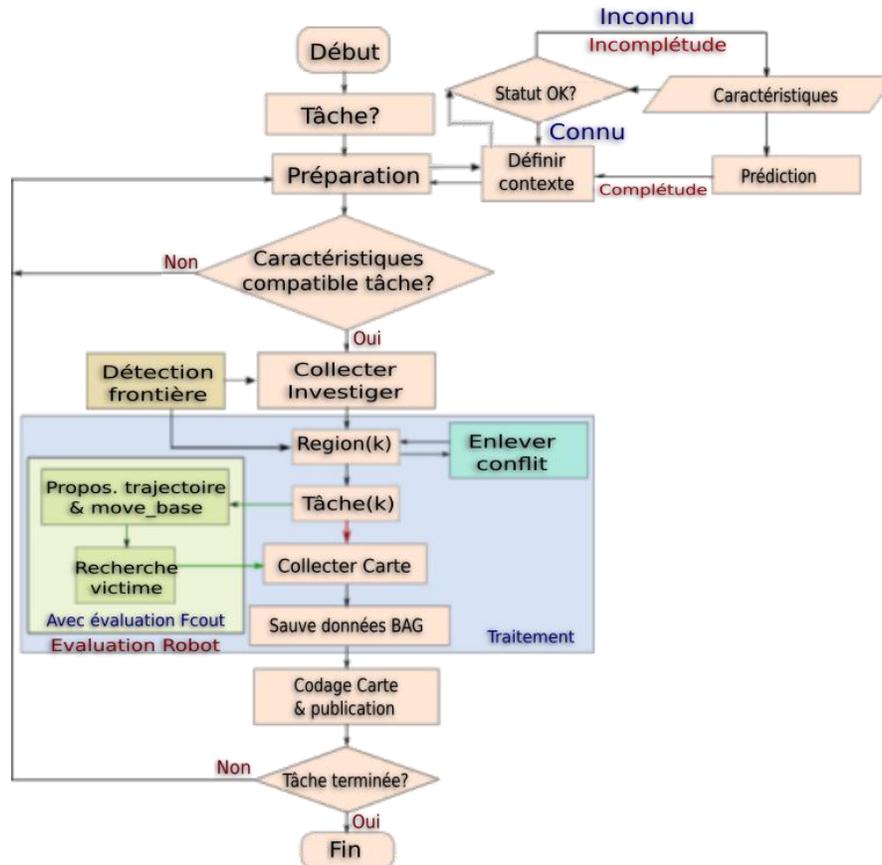


Fig.VI.11 : Organigramme de notre approche

Le cluster collecte des données liées au contexte du ROI, étudie les frontières autour de chaque microrobot, génère des fragments de carte et les partage au sein du cluster. Si les tâches sont terminées, les processus peuvent être éliminés, sinon l'étape de préparation est répétée. Dans un scénario pratique, chaque microrobot doit remplir la condition optimale définie par la fonction de coût f_{cost}^R (Equ.VI.9).

- Connectivité radio au puits ou à l'ensemble du cluster $F_{con/AP}$ qui garantit l'échange de données et/ou maintient un lien (AP pour Access Point).
- Autonomie économique et énergétique F_{auton} quantifie la durée de vie de la batterie pour une tâche particulière.
- $Disp^{all}$ de disponibilité absolue, indique « l'état OK » de l'unité au sein du cluster, signifiant sa capacité à exécuter la tâche avec compétence.

- La disponibilité relative, notée $Disp^{Res}$, signifie l'accessibilité de la charge utile, qui est une ressource spécifique cruciale pour une tâche particulière dont.

$$f_{cost}^{\mu R} = k_1 * F_{con/AP} + k_2 * F_{auton} + k_3 * Disp^{All} + k_4 * Disp^{Res} \quad (Equ.VI.9)$$

avec $k_1+k_2+k_3+k_4=1$

Les poids k_i pour $i \in [1,4]$ symbolisent les contributions de chaque partie spécifique au coût fonctionnel de l'agent, tel que défini précédemment. Notre objectif est de découvrir le résultat optimal (Equ.VI.10) pour la fonction f_{cost} pour chaque agent du cluster, représentant la meilleure solution. On obtient alors l'équation suivante :

$$F_{cost}^{opt} = Max_{j \in [1,R]} \{f_{cost}^{\mu R_j}, constraints\} \quad (Equ.VI.10)$$

L'estimation optimale du cluster est indiquée par sa capacité à déclarer un statut « OK » pour la disponibilité, dépassant la condition minimale requise pour l'exécution des tâches assignées. Nous introduisons $F_{con/AP}$ comme la disponibilité effective réduite d'au moins un lien avec un point d'accès prédéfini. Si F_{con}^{ref} représente le seuil de référence pour une connexion WiFi à un point d'accès et $F_{con}(t)$ signifie le niveau de connectivité actuel d'un agent au point d'accès, alors $F_{con/AP}$ est défini comme le rapport de la connectivité instantanée $F_{con}(t)$ au seuil de référence F_{con}^{ref} . Ceci s'exprime ainsi :

$$F_{con/AP} = 100 * (F_{con}(t)/F_{con}^{ref}) \quad (Equ.VI.11)$$

Par conséquent, F_{auton} signifie l'autonomie de la batterie, indiquant l'énergie restante nécessaire à la batterie pour fournir une puissance adéquate à l'agent pour accomplir avec succès la tâche assignée en tant qu'unité singulière au sein du cluster. Ici, nous représentons τ comme le temps estimé requis pour la tâche assignée, I comme le courant déchargé réel de la batterie en ampère (A) et C comme la capacité de la batterie, agissant comme une source de courant pendant une durée spécifiée en ampère-heure (Ah) et P_{charge} est la puissance de charges consommée par les ressources de l'agent. Les expressions sont donc les suivantes :

$$F_{auton} = C/\tau * (1/I - 10/P_{charge}) \quad (Equ.VI.12)$$

Lors de l'estimation de $Disp^{all}$ et $Disp^{Res}$, la disponibilité d'un membre du cluster est évaluée en fonction de son adéquation effective à une tâche donnée. La disponibilité relative d'une ressource indique la capacité de l'agent à disposer de la ressource spécifiée (charge utile) prête à

l'heure indiquée lorsqu'une tâche est assignée. D'autre part, la disponibilité absolue englobe la somme de toutes les disponibilités relatives (des ressources N_{res}), signifiant que toutes les ressources de l'agent sont prêtes à être utilisées. La disponibilité relative est définie sur 1 lorsqu'il y a un retour positif lors de l'interrogation de la ressource et sur 0 sinon. Donc :

$$D_{Rel/Res} = \begin{cases} 1 & \text{if the targeted resource is available} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{Equ.VI.13})$$

La disponibilité absolue est donnée par :

$$Disp^{All} = \bigcap_{i=1}^{N_{res}} (D_{Rel/Res}^i) \quad (\text{Equ.VI.14})$$

La capacité de l'agent à intervenir pleinement dans une tâche dépend de la connexion ET logique de toutes les fonctions de disponibilité pertinentes. Les coefficients de pondération k_i sont sélectionnés selon différentes méthodes pour optimiser le résultat final de la fonction objectif. Nous adoptons une approche bio-inspirée pour déterminer ces coefficients. La complexité mathématique de chaque système microrobot dépend de ses conditions et exigences spécifiques et est classée comme NP (temps polynomial non déterministe). Le coefficient k_i est déterminé par la significativité du terme multiplicateur utilisé. Chaque multiplicateur est généralement une fonction dépendant d'un composant/ressource spécifique du microrobot et possède ses propres caractéristiques uniques. Le tableau ci-dessous fournit des informations sur les corrélations entre les objectifs de gestion d'un microrobot. Si la fonction d'autonomie énergétique est critique et prioritaire, la capacité d'effectuer une opération de scanning Lidar est également nécessaire pour mener à bien une tâche, même si une autonomie énergétique est disponible. Les quatre fonctions de ressources sont interdépendantes, ce qui ajoute à la complexité de la sélection des valeurs k_i (Tab.VI.1).

Tab.VI.1 : Dépendance mutuelle des fonctions membres

Dépendance	F_{auton}	$Disp^{All}$	$Disp^{Res}$	$F_{con/AP}$
F_{auton}	-	non	oui	oui
$Disp^{All}$	oui	-	oui	oui
$Disp^{Res}$	oui	oui	-	peut être non
$F_{con/AP}$	oui	non	oui	-

Pour déterminer les valeurs optimales des coefficients à l'aide de l'IA, nous pouvons utiliser des algorithmes d'apprentissage automatique tels que la régression, les arbres de décision ou les

réseaux de neurones artificiels. L'idée est d'utiliser un grand ensemble de données comprenant des entrées avec leurs valeurs correspondantes telles que F_{auton} , $Disp^{All}$, $Disp^{Res}$ et $F_{con/AP}$ pour entraîner le modèle. Le modèle peut ensuite être utilisé pour faire des prédictions de chaque valeur en fonction de nouvelles entrées. Les valeurs optimales des coefficients (k_1 , k_2 , k_3 et k_4) peuvent être trouvées en ajustant les paramètres du modèle pour minimiser la différence entre les valeurs prédites (F_{auton} , $Disp^{All}$, $Disp^{Res}$ et $F_{con/AP}$) et les valeurs réelles. Dans l'ensemble de données de formation.

Il convient de noter que ce problème particulier peut être complexe et nécessiter une quantité considérable de données et de ressources informatiques pour obtenir des résultats précis. En outre, il peut être nécessaire de mener des expériences avec divers algorithmes et méthodes d'ingénierie des fonctionnalités pour déterminer le modèle le plus performant.

Un algorithme génétique (Figure VI.12) est une méthode d'optimisation qui s'inspire du processus de sélection naturelle en biologie [327-330]. Il peut être appliqué pour déterminer les coefficients optimaux dans (Equ.VI.9). Un aperçu fondamental des étapes nécessaires pour résoudre le problème à l'aide d'un algorithme génétique serait :

1. Initialiser une population de solutions candidates, chacune représentée par un ensemble de coefficients $\{k_1, k_2, k_3, k_4\}$.
2. Évaluer l'adéquation (P_R) de chaque solution candidate en utilisant les coefficients pour calculer F_{cost} pour chaque exemple dans les ensembles de données de formation. La forme physique est une mesure de l'adéquation de la solution candidate aux données d'entraînement. L'algorithme calcule l'adéquation d'une solution candidate, qui est définie comme l'erreur quadratique moyenne entre les valeurs prédites et les valeurs réelles des fonctions paramètres F_{auton} , $Disp^{All}$, $Disp^{Res}$ et $F_{con/AP}$ (Equ.VI.15).

$$P_R = Mean\left(\left([k_1^*] * F_{con/AP} + [k_2^*] * F_{auton} + [k_3^*] * Disp^{All} + [k_4^*] * Disp^{Res} - [F_{cost}^*]\right)^2\right) \quad (Equ.VI.15)$$

Où $[k_1^*]$, $[k_2^*]$, $[k_3^*]$, $[k_4^*]$ et $[F_{cost}^*]$ sont des vecteurs 2D des valeurs prédites.

3. Sélection des solutions les plus performantes (Une solution candidate la plus performante, également appelée individu ou chromosome, est un ensemble de valeurs pour les coefficients k_1 , k_2 , k_3 et k_4 . Chaque solution candidate est évaluée à l'aide du fonction fitness pour déterminer sa qualité) pour servir de parents à la prochaine

génération. La meilleure solution candidate est déterminée en sélectionnant celle ayant la valeur de fitness la plus faible.

4. Application d'un opérateur génétique de croisement pour combiner deux solutions parentales candidates afin de générer une solution enfant.
5. Pour rendre notre AG adaptatif, en appliquant des opérateurs génétiques de mutations adaptatives qui introduisent des variations dans la population (prend une solution candidate et applique une modification à chaque coefficient). Ajuster le taux de mutation en fonction de la qualité de la population actuelle (dans notre algorithme, le taux de mutation est augmenté si la forme physique moyenne de la population est moins bonne que la meilleure forme physique, et diminué si la forme physique moyenne est meilleure que la meilleure forme physique). Tout d'abord, effectuez un changement de gène aléatoire en utilisant le taux de mutation adapté, puis conservez les paramètres inchangés pendant un intervalle de temps s'il existe pour faire de son gène coefficient un trait récessif en diminuant la valeur. Créez un gène coefficient comme trait dominant du paramètre le plus modifié. Ce qui change beaucoup est controversé et la constante ne nous aide pas à montrer un problème, le cas échéant. Et enfin, quelque temps après le lancement de la mission (après avoir collecté une quantité suffisante d'informations) et grâce à l'IA, nous faisons des prédictions de modifications de chaque gène de coefficient en fonction de son importance lors de l'exécution de la mission.
6. Évaluer l'aptitude de la progéniture et remplacer les solutions les moins performantes de la population par la progéniture.
7. Répétez les étapes 3 à 6 pour un nombre spécifié de générations ou jusqu'à ce qu'un critère d'arrêt soit rempli.
8. Renvoyez la solution la plus performante comme résultat après avoir normalisé les coefficients.

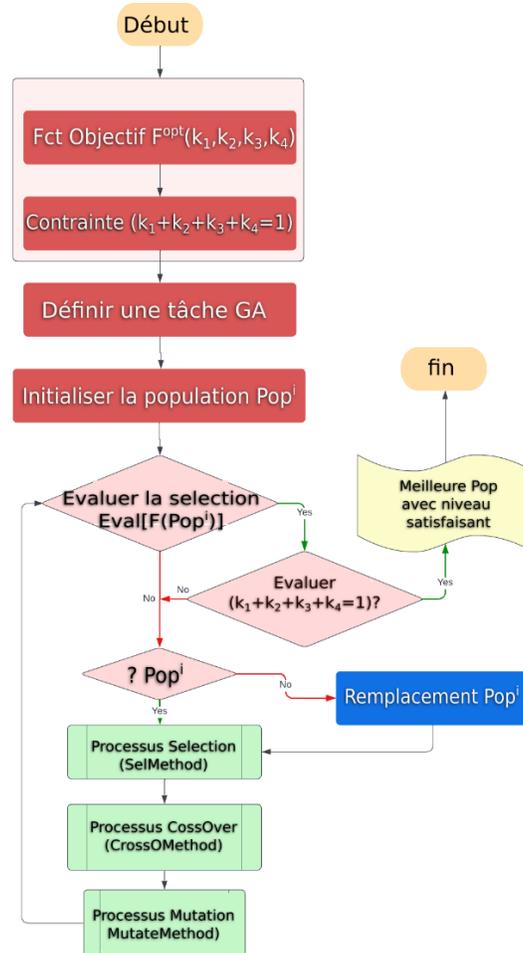


Fig. VI.12 : Algorithme génétique équivalent

Il existe plusieurs façons de mettre en œuvre l'algorithme génétique et différentes optimisations peuvent être utilisées, telles que l'élitisme, la sélection de tournois et la mutation adaptative. Le choix des paramètres, tels que la taille de la population, le taux de mutation et le nombre de générations, est également crucial pour obtenir des performances optimales. De plus, d'autres techniques peuvent offrir différents avantages en termes de calcul, de portabilité et d'efficacité. Pour notre cas d'utilisation spécifique, nous avons décidé d'utiliser des techniques GA.

VI.7. Collaboration par algorithme Greedy pour le MRTA distribué

Nous introduisons un algorithme glouton (Algorithme VI.1) [325] conçu pour le problème MRTA avec messagerie diffusée. Cet algorithme est configuré pour attribuer des agents en fonction de l'affectation optimale à une tâche qui maximise le rapport performance/coût (V_i^k/d_i^k). Ici, V_i^k représente la performance du $k^{\text{ème}}$ agent pour l'accomplissement de la $i^{\text{ème}}$ tâche, et d_i^k

désigne la distance euclidienne entre l'agent et la cible/l'emplacement assigné (Equ.VI.16 et Equ.VI.17).

$$d_i^k = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2} \quad (\text{Equ.VI.16})$$

$$T_i^k = \max_{i \in T} (V_i^k / d_i^k) = \max_{i \in T} (V_i^k \eta_i^k) \text{ avec } \eta_i^k = 1/d_i^k \quad (\text{Equ.VI.17})$$

Ici, la tâche i est la tâche choisie pour le $k^{\text{ème}}$ agent parmi l'ensemble des tâches potentielles, et T représente l'ensemble des tâches viables dans le cadre du $k^{\text{ème}}$ agent sur le total de M tâches disponibles.

L'algorithme, identifié comme algorithme 1, fonctionne comme suit : Dans un premier temps, nous définissons les variables de surface explorées par chaque $robot_k$, où $S_k|_{k \in R}$ désigne l'ensemble des points limites de la surface actuellement balayée. De plus, $f_i^k|_k$ représente une fonction objective (fonction de coût) pour chaque $robot_k$, $task_i$ étant la sous-mission stockée dans T_i^k . De plus, T_i^k c'est l'ensemble des tâches qui n'ont pas encore été assignées et qui sont incluses dans T , représentant le total des tâches disponibles ($T_i^k \in T$). Parallèlement, T^k for $k \in R$ est la tâche sélectionnée pour le robot et P_i^k désigne les emplacements déjà assemblés. Ici, k indique le robot qui doit mettre à jour son offre au stade actuel.

Au départ, aucune tâche n'est assignée, donc $T^k = \emptyset$ pour tous les robots $k \in R$. À chaque étape, une tâche est attribuée indépendamment à un seul robot (Figure VI.13), suite à la décentralisation. Ainsi, nous avons besoin de $|T|$ étapes, le nombre de tâches que le robot doit effectuer pour terminer son état. A chaque itération i , après avoir supprimé les parties en conflit avec les zones explorées par d'autres robots, tous les robots $k \in R$ soumettent une offre (t_i^k, T_i^k). Chaque robot k sélectionne la tâche T^k dans la liste des tâches non localisées T_i^k pour maximiser sa fonction objectif f_i^k . En rassemblant toutes les offres, nous découvrons un gain optimal supérieur pour l'objectif collectif : le succès multiplicatif du groupe F (Equ.VI.18). Grâce aux enchères, nous choisissons efficacement la combinaison paire de tâches-robot optimale pour le plus grand bénéfice global [305, 306].

$$F = Best_{\{T^k\}_{k \in R}} \prod_{k \in R} [f_k(T^k)] \quad (\text{Equ.VI.18})$$

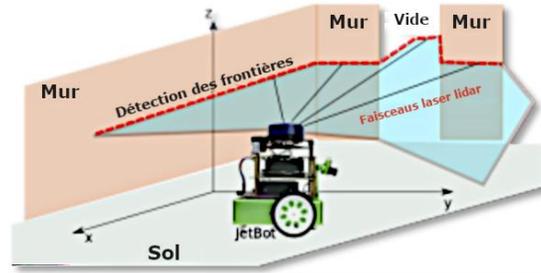


Fig.VI.13 : JetBot pendant l'exploration du domaine

Algorithme VI.1 : La MRTA, Algorithme de recherche orienté glouton

```

Begin
  Initializations:
    Let  $S^k \leftarrow \emptyset$ 
    Let  $K_0 \leftarrow k'$        $\{\forall k' \in [1;R], R: \text{nombre de robots}\}$ 
    Let  $T^k \leftarrow \emptyset$ 
    Let  $P_i^k = \begin{cases} (P_i^k)^* & \text{get all previous tasks if they exist} \\ \emptyset & \text{otherwise} \end{cases}$ 
  for  $(k \in [1; R])$  do
     $S^k = \text{area}(k)$ 
    if  $(k \neq K_0)$  then
      Let  $S^{K_0} \leftarrow (S^{K_0} - S^{K_0} \cap S^k)$ 
    end if
  end for
   $T \leftarrow \text{task}(K_0)_{|S^{K_0}}$ 
   $i \leftarrow 1$ 
  while  $(i \leq |T|)$  do
     $t_i^{K_0} \leftarrow \text{Eval}(f_i^{K_0} |_{S^{K_0}})$        $\{\text{la } i^{\text{th}} \text{ fonction objective}\}$ 
    if  $(T_i^{K_0}(t_i^{K_0}) \subset P_i^{K_0})$  then
       $t_i^{K_0} \leftarrow \text{None}$ 
    end if
     $i \leftarrow i+1$ 
  end while
   $T^{K_0} \leftarrow T_j^{K_0} |_{S^{K_0}, \text{Best}(t_j^{K_0})}$        $\{\text{pour } j \in [1;m], m \text{ est un ensemble de tâches spécifiques}\}$ 
  if  $(T^{K_0} \neq \emptyset)$  then
     $P_i^{K_0} \leftarrow P_i^{K_0} + T^{K_0}$ 
  else
     $T \leftarrow \emptyset$ 
    for  $(k \in [1; R])$  do
      if  $(k \neq K_0)$  do
         $T \leftarrow T + \text{task}(k)_{|S^k, k \neq K_0}$ 
         $i \leftarrow 1$ 
        while  $(i \leq |T|)$  do
           $t_i^k \leftarrow \text{Eval}(f_i^k |_{S^k, k \neq K_0})$ 
          if  $(T_i^k(t_i^k) \subset P_i^k)$  then
             $t_i^k \leftarrow \text{None}$ 
          end if
           $i \leftarrow i+1$ 
        end while
         $T^k \leftarrow T_j^k |_{S^k, \text{Best}(t_j^k), k \neq K_0}$        $\{\text{pour } j \in [1;m] m \text{ est un ensemble de tâches spécifiques}\}$ 
      end if
    end for
     $T^{K_0} \leftarrow \text{Best}(T^k)_{|k \in \{1,2,\dots,R\}}$ 
     $P_i^k \leftarrow P_i^k + T^{K_0} |_{k \in \{1,2,\dots,R\}}$ 
  end if
End.
    
```

Lorsqu'un robot est entouré d'autres robots, il peut être induit en erreur sur ses capacités, entravant ainsi son développement et son exploration. Pour résoudre ce problème et maintenir des performances élevées même lorsqu'il est entouré ($T^k=\emptyset$), notre objectif est d'améliorer la fonction de coût du robot en utilisant les espaces disponibles des robots voisins jusqu'à ce qu'il puisse fonctionner de manière indépendante.

VI.8. Spécifications adoptées du système d'exploitation et du poste de travail

Pour simuler la collaboration et explorer pour localiser une victime, un poste de travail est utilisé. Cette station de travail comprend un HP ProBook x360 435 G7 avec un processeur AMD Ryzen 7 PRO 4750U, une carte graphique AMD Radeon RX Vega 7, 32 Go de RAM DDR4 Kingston et un disque SSD Samsung de 1 To. La simulation utilise ROS [331] sur Linux Ubuntu 20.04, améliorant la communication entre les nœuds et facilitant la transmission des messages entre les robots et les ordinateurs. De plus, JetBot (Fig .VI.14), optimisé par NVIDIA Jetson Nano et accessible chez NVIDIA, prend en charge les capteurs et implémente RNN pour la détection d'objets et l'évitement des collisions. De plus, JetBot est capable de se connecter via diverses liaisons radio.



Fig.VI.14 : JetBot de Nvidia

VI.9. Scenarios et résultats

Le scénario de simulation est illustré dans la figure VI.15. Il est caractérisé par diverses contraintes qui évaluent l'efficacité et la fiabilité de notre approche pour naviguer dans un cluster de microrobots, avec et sans collaboration.

La scène de simulation est constituée de pièces (espaces isolés) contenant des meubles (obstacles). Les microrobots travaillent ensemble pour sauver plusieurs victimes, afin de minimiser le temps et d'améliorer la fiabilité.



Fig.VI.15 : Zone modèle de la simulation

La simulation illustre trois niveaux de gravité impliquant 1 à 3 microrobots (Figure VI.16).

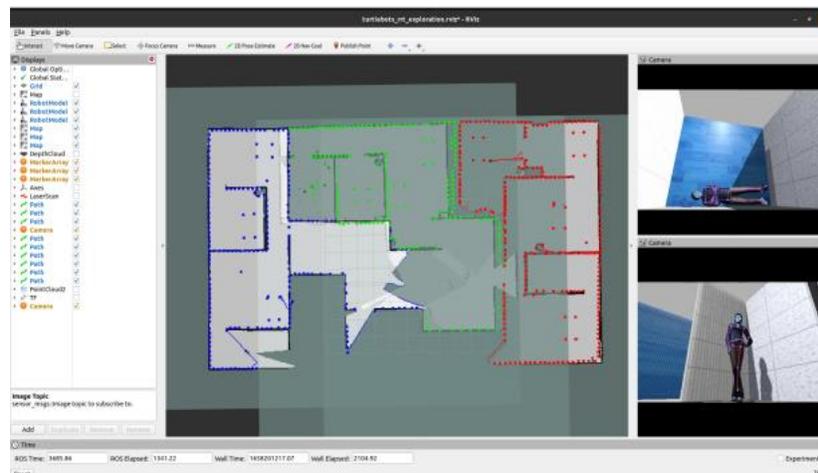


Fig.VI.16 : Segmentation spatiale

Nous avons fait les hypothèses suivantes pour la simulation :

- Le domaine et les dimensions de la scène restent constants tout au long de l'enquête ($\approx 25 \times 15$ mètres).
- La position cible dans la scène n'affecte pas de manière significative la durée de la simulation (on a fait trois expérimentations de positionnement des cibles dans la scène de mission).
- Les agents (microrobots) sont supposés homogènes, possédant des caractéristiques identiques (Figures VI.17 et VI.18).
- Les conditions de connectivité, de charge utile et d'autonomie sont satisfaisantes pour toutes les unités du cluster.

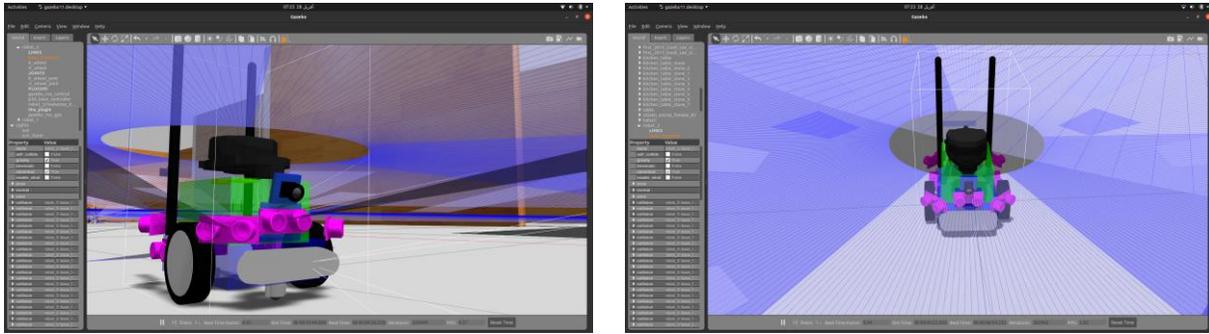


Fig.VI.17 : Le modèle de robot utilisé (JetBot)

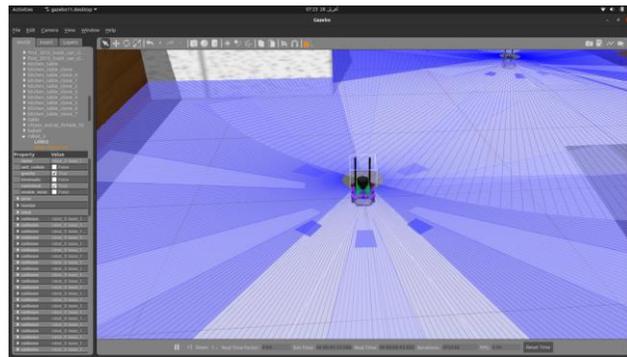


Fig.VI.18 : Robots lors de l'exécution d'une mission

Les résultats de la simulation sont présentés dans les figures et tableaux suivants (figures VI.19 à VI.41 et tableaux VI.2 à VI.8). Les régions de la figure VI.19 montrent les zones désignées en couleurs RVB au sein de la zone entière. Ces domaines proviennent de la segmentation du retour sur investissement total à l'aide de l'algorithme gourmand. Le processus est répété trois fois.

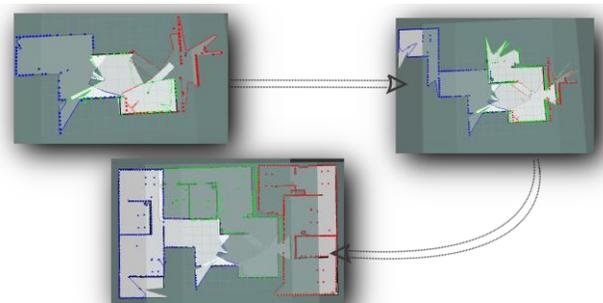


Fig.VI.19 : Mises à jour du ROI des agents

Le processus implique dix simulations, chacune effectuée trois fois. Les résultats sont évalués selon cinq scénarios différents (cas I à V). Dans chaque test, on note le temps nécessaire pour atteindre tous les objectifs jusqu'à trouver deux cibles (victimes), avec et sans considération de collaboration.

- Scénario I (One Robot) : robot unique (voir le tableau VI.2).

- Scénario II (2NoCollab) : Deux robots sans algorithme de collaboration (voir le tableau VI.3).
- Scénario III (2WCollab) : Deux robots avec algorithme de collaboration (voir le tableau VI.4).
- Scénario IV (3NoCollab) : Trois robots sans algorithme de collaboration (voir Tableau VI.5).
- Scénario V (3WCollab) : Trois robots avec algorithme de collaboration (voir le tableau VI.6).

- Premier cas (One Robot) :

Le cas se devise en trois expérimentations d'emplacement des cibles comme le montre les trois captures suivantes (Figure VI.20), la scène de simulations est répétée dix fois pour chaque expérimentation d'emplacement des cibles.



Fig.VI.20 : Scènes avec emplacements robot-cibles

Où la répartition temporelle par nombre de tentatives est donnée (avec les lignes de la moyenne statistique) dans les graphiques suivant de chaque expérimentation d'emplacement des cibles (Figures VI.21, VI.22 et VI.23).

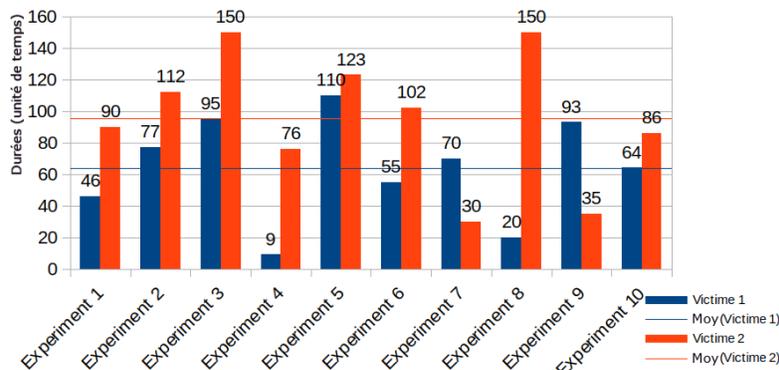


Fig.VI.21 : Temps mis par un seul robot pour atteindre les cibles (1^{er} emplacement des cibles)

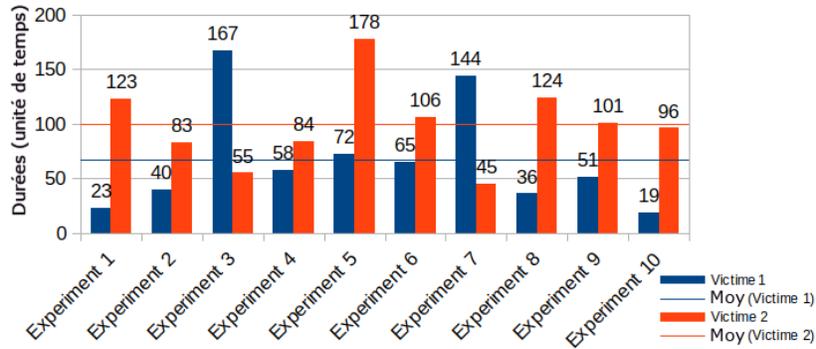


Fig.VI.22 : Temps mis par un seul robot pour atteindre les cibles (2^{ème} emplacement des cibles)

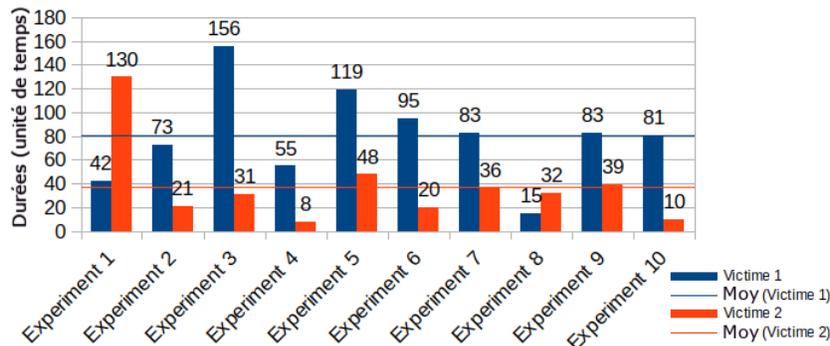


Fig.VI.23 : Temps mis par un seul robot pour atteindre les cibles (3^{ème} emplacement des cibles)

- Deuxième cas (2NoCollab) :

Le cas se devise en trois expérimentations d’emplacement des cibles comme le montre les trois captures suivantes (Figure VI.24), la scène de simulations est répétée dix fois pour chaque expérimentation d’emplacement des cibles.



Fig.VI.24 : Scènes avec emplacement robots-cibles

Où la répartition temporelle par nombre de tentatives est donnée (avec les lignes de la moyenne statistique) dans les graphiques suivant de chaque expérimentation d’emplacement des cibles (Figures VI.25, VI.26 et VI.27).

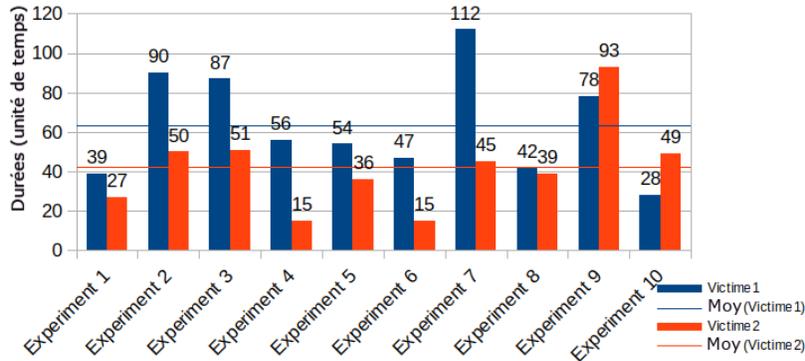


Fig.VI.25 : Temps mis par les deux robots pour atteindre les cibles (1^{er} emplacement des cibles)

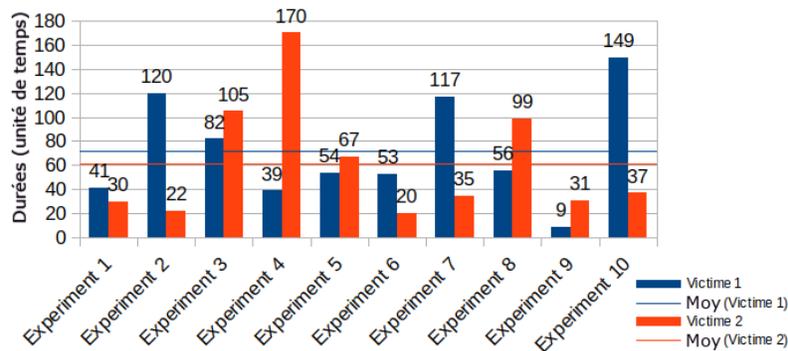


Fig.VI.26 : Temps mis par les deux robots pour atteindre les cibles (2^{ème} emplacement des cibles)

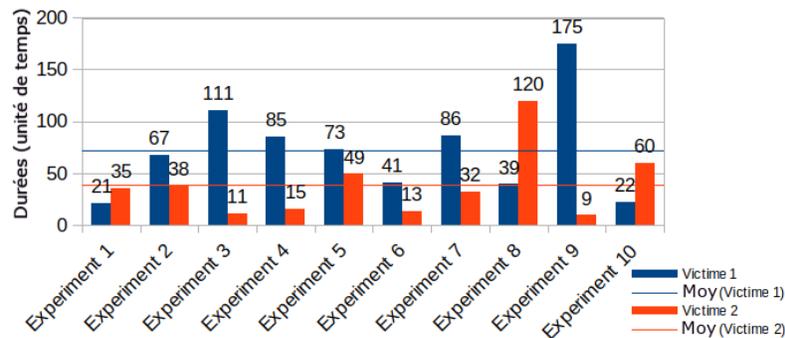


Fig.VI.27 : Temps mis par les deux robots pour atteindre les cibles (3^{ème} emplacement des cibles)

- Troisième cas (2WCollab) :

Le cas se devise en trois expérimentations d’emplacement des cibles comme le montre les trois captures suivantes (Figure VI.28), la scène de simulations est répétée dix fois pour chaque expérimentation d’emplacement des cibles.



Fig.V.28 : Scènes avec emplacement robots-cibles

Où la répartition temporelle par nombre de tentatives est donnée (avec les lignes de la moyenne statistique) dans les graphiques suivant de chaque expérimentation d'emplacement des cibles (Figures VI.29, VI.30 et VI.31).

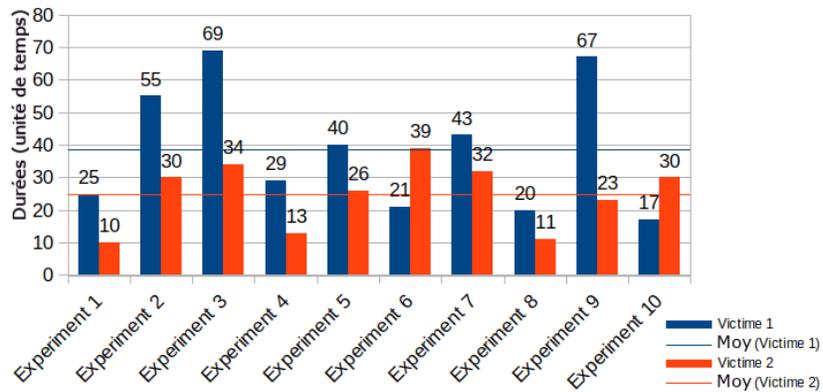


Fig.VI.29 : Temps mis par les deux robots pour atteindre les cibles (1^{er} emplacement des cibles)

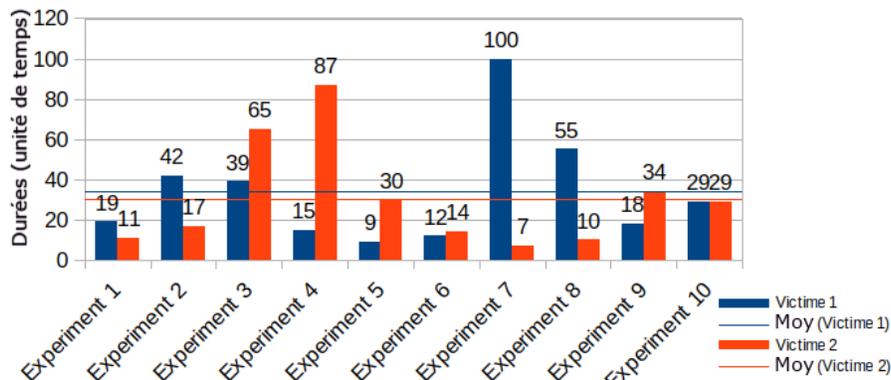


Fig.VI.30 : Temps mis par les deux robots pour atteindre les cibles (2^{dème} emplacement des cibles)

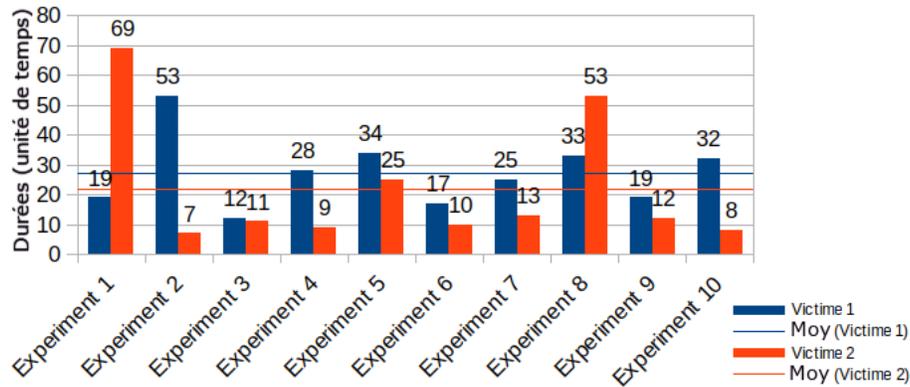


Fig.VI.31 : Temps mis par les deux robots pour atteindre les cibles (3^{ème} emplacement des cibles)

- Quatrième cas (3NoCollab) :

Le cas se divise en trois expérimentations d’emplacement des cibles comme le montre les trois captures suivantes (Figure VI.32), la scène de simulations est répétée dix fois pour chaque expérimentation d’emplacement des cibles.



Fig.VI.32 : Scènes avec emplacement robots-cibles

Où la répartition temporelle par nombre de tentatives est donnée (avec les lignes de la moyenne statistique) dans les graphiques suivant de chaque expérimentation d’emplacement des cibles (Figures VI.33, VI.34 et VI.35).

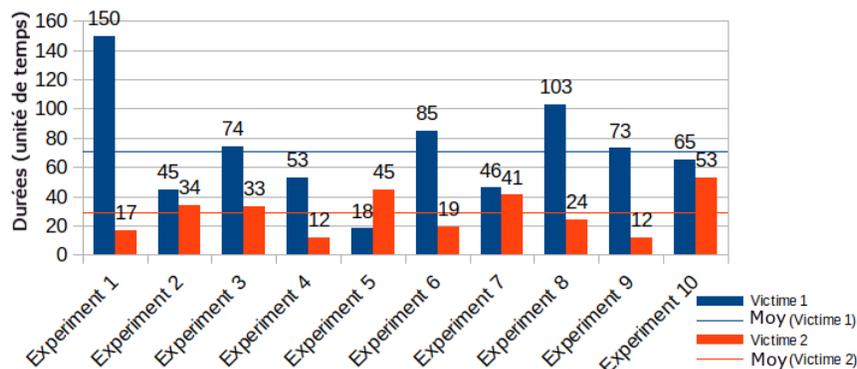


Fig.VI.33 : Temps mis par les trois robots pour atteindre les cibles (1^{er} emplacement des cibles)

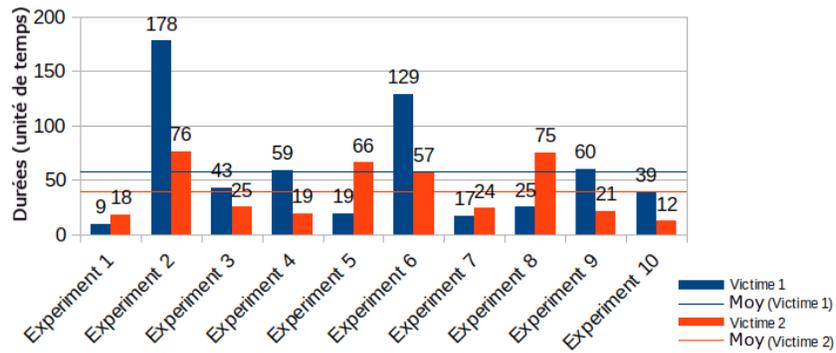


Fig.VI.34 : Temps mis par les trois robots pour atteindre les cibles (2^{ème} emplacement des cibles)

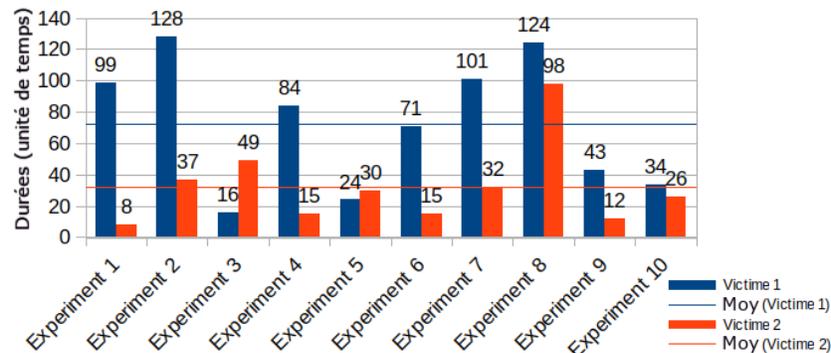


Fig.VI.35 : Temps mis par les trois robots pour atteindre les cibles (3^{ème} emplacement des cibles)

- Cinquième cas (3WCollab) :

Le cas se divise en trois expérimentations d'emplacement des cibles comme le montre les trois captures suivantes (Figure VI.36), la scène de simulations est répétée dix fois pour chaque expérimentation d'emplacement des cibles.



Fig.VI.36 : Scènes avec emplacement robots-cibles

Où la répartition temporelle par nombre de tentatives est donnée (avec les lignes de la moyenne statistique) dans les graphiques suivant de chaque expérimentation d'emplacement des cibles (Figures VI.37, VI.38 et VI.39).

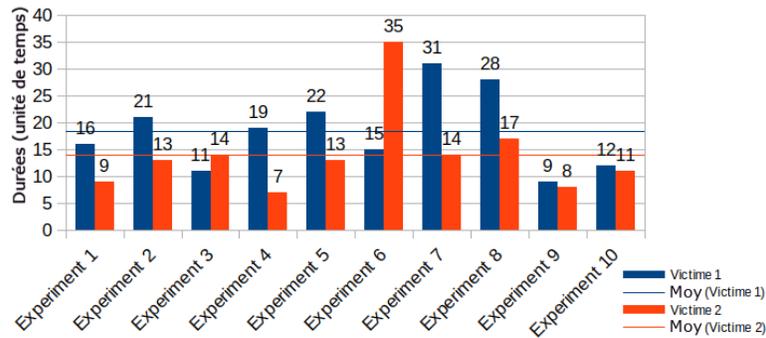


Fig.VI.37 : Temps mis par les trois robots pour atteindre les cibles (1^{er} emplacement des cibles)

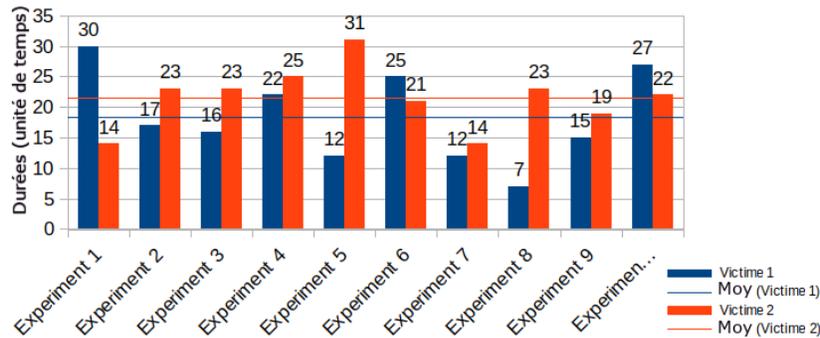


Fig.VI.38 : Temps mis par les trois robots pour atteindre les cibles (2^{ème} emplacement des cibles)

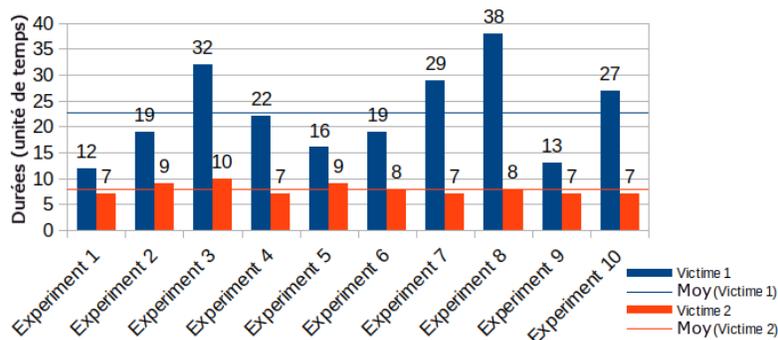


Fig.VI.39 : Temps mis par les trois robots pour atteindre les cibles (3^{ème} emplacement des cibles)

VI.10. Discussion et perspectives

Le plan consiste à trouver le temps nécessaire pour atteindre chaque victime, à identifier le temps le plus long pour atteindre deux victimes et à faire la moyenne des valeurs maximales. Nous nous concentrons sur les durées moyennes maximales et minimales (Tableaux VI.2 à VI.7).

Tab. VI.2 : Temps de simulation d'un robot pour retrouver les deux victimes (en unité de temps)

	Cible 1	Cible 2	Durée maximale
Emplacement 1	63.9	95.4	105.2
Emplacement 2	67.5	99.5	120.6
Emplacement 3	80.2	37.5	90.7

Tab. VI.3 : Durées moyennes de deux robots sans algorithme de collaboration (en unité de temps)

	Cible 1	Cible 2	Durée maximale
Emplacement 1	62.4	40.3	66.9
Emplacement 2	72	61.6	95.2
Emplacement 3	72	38.2	85.3

Tab. VI.4 : Durées moyennes de deux robots avec algorithme de collaboration (en unité de temps)

	Cible 1	Cible 2	Durée maximale
Emplacement 1	60.1	38.9	41.7
Emplacement 2	33.8	30.4	47.5
Emplacement 3	27.2	21.7	34.2

Tab. VI.5 : Durées moyennes de trois robots sans algorithme de collaboration (en unité de temps)

	Cible 1	Cible 2	Durée maximale
Emplacement 1	71.2	29	73.9
Emplacement 2	57.8	39.3	69.1
Emplacement 3	72.6	32.2	76.5

Tab. VI.6 : Durées moyennes de trois robots avec algorithme de collaboration (en unité de temps)

	Cible 1	Cible 2	Durée maximale
Emplacement 1	18.4	14.1	20.7
Emplacement 2	18.3	21.5	24

Emplacement 3	22.7	7.9	22.7
----------------------	------	-----	------

Tab.VI.7 : Durées moyennes par expérience (en unité de temps)

	Emplacement 1	Emplacement 2	Emplacement 3
One Robot	90.7	120.6	105.2
2NoCollab	85.3	95.2	66
2WCollab	34.2	47.5	66
3NoCollab	76.5	69.1	73.9
3WCollab	22.7	24	20.7

Les résultats globaux sont compilés sous forme de valeurs minimales (l'identification de la première cible) et maximales (l'identification de la deuxième cible ou la fin de la mission) pour une consolidation potentielle (Tableau VI.8).

Tab.VI.8 : Temps de simulation pour retrouver les deux victimes (en unité de temps)

	One	2NoCol	2WCol	3NoCol	3WCol
Min	90.7	66.9	34.2	69.1	20.7
Max	120.6	95.2	47.5	76.5	24

La collaboration a un impact de plus en plus long sur la durée (Figures VI.40 et VI.41). De plus, à mesure que le nombre d'agents augmente, la durée diminue, en particulier lorsqu'une stratégie de collaboration est activée.

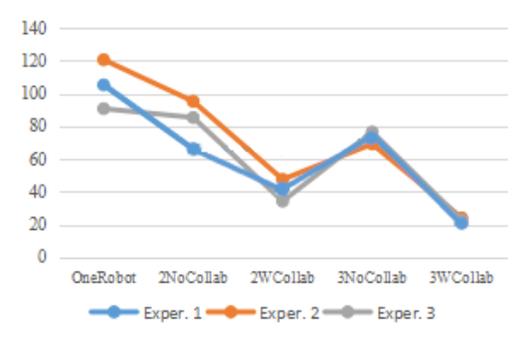


Fig.VI.40 : Progression de la durée au fil de l'expérimentation

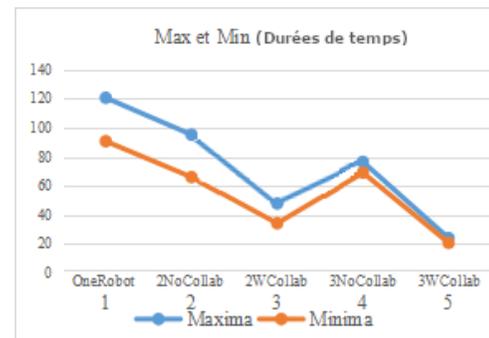


Fig.VI.41 : Durée maximale et minimale par expérience jusqu'à ce que la victime soit trouvée

Le retour sur investissement a été évalué de deux manières : en localisant les cibles et en scannant la zone en 3D pour une reconnaissance potentielle. Cela a été réalisé en utilisant une version améliorée d'un Lidar horizontal (RPLiDAR A1M8) dans le cadre des nouvelles fonctionnalités de notre plateforme AL Moustaksheef3D, une charge utile robotique pour le SLAM 3D actuellement en développement (Figure VI.42). Le Lidar amélioré (Figure VI.43) a été testé

sur un engin à roues et un drone. Les données collectées (Figure VI.44) sont fournies au format PCD. Ce modèle intègre deux Lidar (un réel et autre logique), le premier étant utilisé pour la géolocalisation (Gmapping puis navigation) et le second pour la vision 3D. La reproduction de la scène produit des données relatives pour estimer les faits dans la zone d'étude. Les données de nuages de points sont utilisées pour la reconstruction de la scène. Des logiciels tels que CloudCompare, MeshLab, Blender et Gimp permettent le traitement en arrière-plan des données de nuages de points. Les agents micro-cobot intègrent des API dans leur logique embarquée, employant des algorithmes robustes pour identifier des cibles spécifiques dans des environnements hostiles. En utilisant une approche MRTA avec l'algorithme Greedy amélioré, notre stratégie optimise l'identification des cibles grâce à la coopération d'agents de cluster, réduisant considérablement le temps de tâche. Cette approche collaborative nécessite des capacités de processeur substantielles, avec des calculs de base centralisés pour la collaboration et décentralisés pour la prise de décision indépendante des agents après l'attribution des tâches.

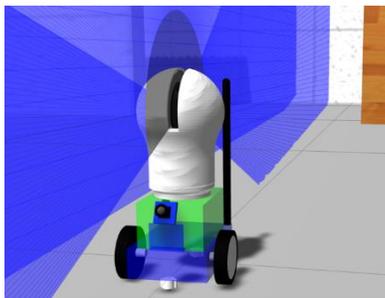


Fig.VI.42 : JetBot/AlMustakhsheef3D



Fig.VI.43 : Le modèle 3D du nouveau LiDAR

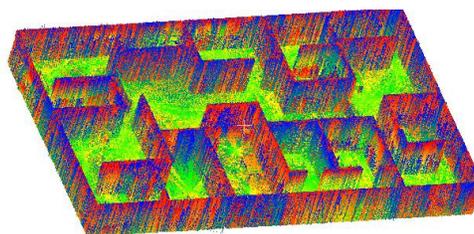


Fig.VI.44 : Segmentation ROI SLAM, sur RVIZ et Gazebo

VI.11. Conclusion

Le projet vise à concevoir des robots collaboratifs dotés de capacités physiques et d'intelligence pour travailler efficacement dans des environnements difficiles. Ces cobots contribueront à réduire la charge de travail des équipes d'intervention humaine ou de secours en identifiant les cibles. Ils

seront également utiles dans les zones où la densité de robots et de techniciens présente un risque pour la sécurité humaine.

Nos recherches se concentrent sur les microrobots collaborant avec les humains, conduisant au concept de micro-cobots. Cela vise à réduire la charge utile et à minimiser les goulots d'étranglement dans les tâches. L'augmentation des agents microrobots améliore la vitesse d'investigation et la recherche de cibles. Les résultats confirment notre réflexion initiale et confirment les hypothèses prédéfinies. Les figures VI.40 et VI.41 montrent que la collaboration améliore considérablement la progression du cluster dans la mission de recherche. La collaboration empêche le comportement erratique des microrobots, réduit la durée et empêche l'effondrement du système. Moins d'agents entraînent davantage de dispersion et de chaos. Un nombre modéré d'investigateurs ($1 < N_R \leq 3$) est préférable. Des techniques d'essaim avec collaboration sont utilisées pour gérer la situation.

Conclusion générale

Conclusion générale

Durant l'édition de cette thèse nous avons exploré le domaine complexe de la collaboration entre micro-cobots dans des sites dits hostiles. Notre exploration a touché, et présenté de manière complète les défis, les stratégies et les résultats du déploiement de ces entités robotiques à petite échelle pour des opérations de recherche et de sauvetage. Grâce à un examen du cadre théorique, des méthodologies de simulation et des approches et implémentations algorithmiques, cette étude visait à exposer la faisabilité et l'efficacité des micro-cobots pour répondre aux complexités des environnements hostiles, ainsi à la problématique indiquée au début. Notre contribution, n'est qu'une expérience à la suite de beaucoup de travaux de recherche [8, 9, 14, 19, 20, 22, 332, 333].

La mise en œuvre d'algorithmes collaboratifs et de recherches, y compris l'algorithme glouton, les algorithmes génétiques, a démontré la capacité des micro-cobots à travailler en tandem, optimisant les modèles de recherche et l'allocation des ressources au sein des sites hostiles. L'environnement de simulation, conçu pour reproduire les défis du monde réel, a fourni des informations précieuses sur la manière dont les micro-cobots naviguent et collaborent dans des scénarios dynamiques et imprévisibles. L'inclusion de l'exploration dans des sites inconnus et renfermes trop de défis, et qui peut même renfermes des êtres humains, a testé l'adaptabilité des micro-cobots, démontrant leur potentiel à surmonter divers obstacles environnementaux. L'étude s'est penchée sur les protocoles de communication et les stratégies de coordination, révélant l'importance d'un échange transparent d'informations entre les micro-cobots. Une collaboration efficace reposait sur la capacité de ces robots à partager des données, à s'adapter à des conditions changeantes et à atteindre collectivement les objectifs de la mission. L'intégration de Framework et d'outils de simulation a permis une évaluation contrôlée de la collaboration micro-cobot. Les résultats des sites hostiles simulés ouvrent la voie à l'extrapolation des leçons apprises à des scénarios du monde réel, soulignant ainsi le caractère pratique et la pertinence de l'étude. Ces recherches apportent plusieurs contributions notables dans le domaine de la microrobotique et de la robotique collaborative :

- Les algorithmes proposés constituent une base pour améliorer les capacités collaboratives des micro-cobots, en relevant les défis uniques posés par les environnements hostiles.

- L'exploration détaillée des complexités des sites contribue à une compréhension plus approfondie de la façon dont les micro-cobots peuvent naviguer et fonctionner dans des environnements difficiles et dynamiques.
- L'étude valide l'utilisation des simulations en tant qu'outil efficace pour évaluer la collaboration entre micro-cobots, fournissant des informations précieuses qui comblent le fossé entre les modèles théoriques et les applications du monde réel.

Bien que cette recherche ait fourni des informations précieuses, le domaine de la collaboration entre micro-cobots sur des sites hostiles est vaste et en constante évolution. Les futurs efforts de recherche pourraient explorer :

- Fusion et perception de capteurs : amélioration des capacités sensorielles des micro-cobots pour une meilleure perception de l'environnement et une meilleure collecte d'informations.
- Swarm Intelligence : enquête sur les techniques avancées d'intelligence par essaim pour améliorer encore la collaboration, l'adaptabilité et la tolérance aux pannes.
- Interaction homme-robot : examiner l'intégration de micro-cobots avec des intervenants humains dans des scénarios de catastrophe, explorer les moyens d'optimiser la coopération entre les agents robotiques et humains.

En conclusion, les efforts collaboratifs des micro-cobots sur des sites hostiles présentent une frontière prometteuse pour la robotique et la réponse aux catastrophes. Les résultats de cette thèse contribuent au corpus croissant de connaissances dans le domaine, ouvrant la voie à des progrès qui peuvent à terme sauver des vies et améliorer l'efficacité des opérations de recherche et de sauvetage dans des environnements difficiles et imprévisibles. Alors que nous nous dirigeons vers un avenir où la robotique jouera un rôle de plus en plus crucial dans la gestion des catastrophes, les leçons tirées de cette étude servent de tremplin pour une exploration et une innovation continues dans la collaboration micro-cobot.

Les études relevant de ce genre de thématique et se basant sur des stratégies de validation utilisant des outils de simulation, restent toujours à vérifier dans les cas réels, sur terrain et avec des machines microrobots, comme décrit plus haut. Certes, beaucoup de contraintes sont à surmonter. En simulation, le temps s'écoule en unité de temps de simulation (stu) dépendant grandement des références de stations de calcul utilisée, dans la plupart du temps c'est des machines 64bits avec des cartes puissantes graphiques (Cores CPU, Cores GPU, Nombre de

threads, VRAM et caches d'échange). En réalité, ces mêmes machines sont des unités (agent) microrobot dont la morphologie nécessite le montage de microcontrôleurs NVIDIA Jetson Nano sur un châssis et doté d'équipements représentant la charge utile (VLiDAR3D version amélioré du LPLIDAR A1M8), de caméra PiCamera, d'un IMU GY-85 BMP085 Gyromètre-Accéléromètre 9 Axes. Le châssis est motorisé par des moteurs CC commandés par driver. Chaque agent, renferme un ensemble de codes (les bibliothèques et les codes des algorithmes) en action, à travers le système d'exploitation Ubuntu ARM.

L'environnement est bien connu pour sa stabilité logicielle et sa robustesse dans des situations de validation sur le terrain en mode réel et même conjointement avec la simulation. Notre thèse, tente à explorer les aptitudes en mode réel d'un microrobot. Il est certain que le conditionnement mécanique (poids, friction, inertie, ...), électrique (batterie, autonomie, câblage, ...) ou en relation avec les sites réel de validation, surajouteront des défis à solutionner. En perspective, nous développerons un cluster de validation de l'approche, utilisant des microrobots reconfigurables (robot à morphologie reconfigurable) afin d'augmenter la pertinence de la coopération.

Références

Références

- [1] D. Tihanyi, Y. Lu, O. Karaca, and M. Kamgarpour, “Multi-robot task allocation for safe planning against stochastic hazard dynamics,” *2023 European Control Conference, ECC 2023*, 2023, doi: 10.23919/ECC57647.2023.10178126.
- [2] I. Rekleitis, G. Dudek, and E. Miliotis, “Multi-robot collaboration for robust exploration,” *Ann Math Artif Intell*, vol. 31, no. 1–4, pp. 7–40, 2001, doi: 10.1023/A:1016636024246.
- [3] NVIDIA Corporation, “Nvidia Tesla V100 GPU Volta Architecture,” *White Paper*, no. v1.1, 2017.
- [4] Y. Adrian, R. C. Lesmana, and Sudimanto, “Analisis Performa pada Video Graphic Array (VGA) Nvidia GTX 950M DDR3 dan Nvidia GTX 950M GDDR5,” *Media Informatika*, vol. 20, no. 2, 2021, doi: 10.37595/mediainfo.v20i2.74.
- [5] X. Li, “An Analysis of the Future of Nvidia,” *Advances in Economics, Management and Political Sciences*, vol. 57, no. 1, 2024, doi: 10.54254/2754-1169/57/20230728.
- [6] J. Choquette, W. Gandhi, O. Giroux, N. Stam, and R. Krashinsky, “NVIDIA A100 Tensor Core GPU: Performance and Innovation,” *IEEE Micro*, vol. 41, no. 2, 2021, doi: 10.1109/MM.2021.3061394.
- [7] NVIDIA, “NVIDIA TensorRT,” *NVIDIA Developer*, no. June, 2021.
- [8] A. S. Howe and J. W. Howe, “Applying construction automation research to extraterrestrial building projects,” in *SAE Technical Papers*, 2000. doi: 10.4271/2000-01-2465.
- [9] G. L. Serejo, V. A. dos Santos, A. F. B. Silva, and C. G. R. dos Santos, “Mobile Platform Based on ROS and LIDAR for Mapping in Civil Construction,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2023. doi: 10.1007/978-3-031-37105-9_14.
- [10] M. A. Rahman, M. S. Miah, W. Gueaieb, and A. El Saddik, “A framework for sensory-based P2P collaborative environment,” in *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, 2007. doi: 10.1109/imtc.2007.379047.
- [11] A. Khamis, A. Hussein, and A. Elmogy, “Multi-robot Task Allocation: A Review of the State-of-the-Art,” 2015, pp. 31–51. doi: 10.1007/978-3-319-18299-5_2.
- [12] H. Chakraa, F. Guérin, E. Leclercq, and D. Lefebvre, “Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art,” *Robotics and Autonomous Systems*, vol. 168. 2023. doi: 10.1016/j.robot.2023.104492.
- [13] M. Garzón *et al.*, “Using ROS in multi-robot systems: Experiences and lessons learned from real-world field tests,” *Studies in Computational Intelligence*, vol. 707, pp. 449–483, 2017, doi: 10.1007/978-3-319-54927-9_14.
- [14] Y. Yu, Z. Miao, X. Wang, and L. Shen, “Cooperative circumnavigation control of multiple unicycle-type robots with non-identical input constraints,” *IET Control Theory and Applications*, vol. 16, no. 9, pp. 889–901, 2022, doi: 10.1049/cth2.12275.
- [15] I. Tkach and Y. Edan, *Distributed Heterogeneous Multi Sensor Task Allocation Systems*, vol. 7. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-34735-2.
- [16] I. Navarro and F. Matía, “An Introduction to Swarm Robotics,” *ISRN Robotics*, vol. 2013, 2013, doi: 10.5402/2013/608164.

-
- [17] E. Şahin and L. Bayındır, "A Review of Studies in Swarm Robotics," *Turkish Journal of Electrical Engineering and Computer Science*, vol. 15, no. 2, pp. 115–147, 2014, [Online]. Available: <http://dergipark.gov.tr/tbtkelektrik/issue/12085/144468>
- [18] A. Roli, A. Ligot, and M. Birattari, "Complexity Measures: Open Questions and Novel Opportunities in the Automatic Design and Analysis of Robot Swarms," *Front Robot AI*, vol. 6, 2019, doi: 10.3389/frobt.2019.00130.
- [19] T. Abderrahmane, N. Amardjia, and T. Mohammed, "Securing laboratories through internet of things networks: an ocomprehensive approach for ensuring safety and efficiency," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 572–585, Feb. 2024, doi: 10.11591/eei.v13i1.6728.
- [20] A. Tamali, N. Amardjia, and M. Tamali, "Distributed and autonomous multi-robot for task allocation and collaboration using a greedy algorithm and robot operating system platform," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 13, no. 2, p. 205, Jun. 2024, doi: 10.11591/ijra.v13i2.pp205-219.
- [21] A. Giusti, G. C. Maffettone, D. Fiore, M. Coraggio, and M. di Bernardo, "Distributed control for geometric pattern formation of large-scale multirobot systems," *Front Robot AI*, vol. 10, 2023, doi: 10.3389/frobt.2023.1219931.
- [22] P. Ghassemi and S. Chowdhury, "Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints," *Rob Auton Syst*, vol. 147, 2022, doi: 10.1016/j.robot.2021.103905.
- [23] B. P. Gerkey and M. J. Mataric, "Multi-robot task allocation: Analyzing the complexity and optimality of key architectures," *Proc IEEE Int Conf Robot Autom*, vol. 3, pp. 3862–3868, 2003, doi: 10.1109/robot.2003.1242189.
- [24] J. Ferber, "Multi-Agent System: An Introduction to Distributed Artificial Intelligence," 1999.
- [25] H. Rowaihy *et al.*, "A Survey of Sensor Selection Schemes in Wireless Sensor Networks," *Proc SPIE*, vol. 6562, p. 35, 2007, doi: 10.1117/12.723514.
- [26] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," *ICRA workshop on open source system*, 2009.
- [27] G. B. & S. Pesty, "Cognition, Affects et Interaction," *Master Ingénierie de la Cognition, de la Création et des Apprentissages*, 2015.
- [28] P. Goutefangea, "Isaac Asimov : les (quatre) ' lois de la robotique ' et l'échange de paroles," 2016, Accessed: Feb. 05, 2024. [Online]. Available: <https://hal.science/hal-01397948>
- [29] A. M. Laura Buller, Clive Gifford, *ROBOT*. DK Publishing, 2018. doi: 10.16309/j.cnki.issn.1007-1776.2003.03.004.
- [30] R. Mitchell, "The tragedy of 3-D cinema," *Film History: An International Journal*, vol. 16, no. 3, pp. 208–215, Sep. 2004, doi: 10.2979/FIL.2004.16.3.208.
- [31] J.-P. Laumond, D. Vidal, and A. L. Boutin, *Robots*. 2018.
- [32] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, doi: 10.1093/mind/LIX.236.433.
- [33] P. Goutefangea, "Alan Turing, le «jeu de l'imitation» et la première personne," 2017, Accessed: Feb. 05, 2024. [Online]. Available: <https://hal.science/hal-01397948>
- [34] N. pekka and A. Ogbechie, "Industrial and Service Robots," no. 31, 2016.
-

-
- [35] S. G. U. Y. Rocheleau, "Petit robot marcheur: Plateforme tout-terrain (PROMPT)," 2010.
- [36] S. Kawamura *et al.*, "Neural networks IC controlled multi-legged walking MEMS robot with independent leg mechanism," *Artif Life Robot*, vol. 23, no. 3, pp. 380–386, Sep. 2018, doi: 10.1007/s10015-018-0445-y.
- [37] M. Z. Miskin *et al.*, "Electronically integrated, mass-manufactured, microscopic robots," *Nature*, vol. 584, no. 7822, pp. 557–561, Aug. 2020, doi: 10.1038/s41586-020-2626-9.
- [38] A. A. Basheer, "Graphene materials for fabrication of robots," *Mater Chem Phys*, vol. 302, p. 127781, Jul. 2023, doi: 10.1016/j.matchemphys.2023.127781.
- [39] Martin, "Le robot-serpent d'exploration en milieu hostile," [Online], 2016.
- [40] A. Oeillet, "Un robot-cafard plus vrai que nature développé en Russie," [Online], 2015.
- [41] "Le robot de fraisage élimine tous les obstacles dans les canalisations souterraines," [Online], 2013.
- [42] "Les différent type de robot," [Online], 2019.
- [43] A. Lagneau, "L'entreprise israélienne General Robotics Ltd propose un robot armé pour les opérations spéciales," [Online], 2016.
- [44] Mikael. A, "Gard : les robots du démantèlement nucléaire au Visiatome à Marcoule," [Online], 2014.
- [45] CNRS, "Robotique aéro-terrestre," [Online], 2008.
- [46] L. G. Terveen, "Overview of human-computer collaboration," *Knowl Based Syst*, vol. 8, no. 2–3, pp. 67–81, Apr. 1995, doi: 10.1016/0950-7051(95)98369-H.
- [47] B. Claverie, B. L. Blanc, and P. Fouillat, "La cobotique : la robotique soumise," *Communication et organisation*, vol. 44, pp. 203–214, 2013.
- [48] B. Claverie, B. Le Blanc, and P. Fouillat, "La cobotique," *Communication et organisation*, no. 44, pp. 203–214, Dec. 2013, doi: 10.4000/communicationorganisation.4425.
- [49] T. Moulières-Seban, "Conception de systèmes cobotiques industriels : approche cognitive à la production pyrotechnique au sein de Safran et ArianeGroup," ED SPI 209, 2017.
- [50] J. J. Atain-Kouadio and A. Sghaier, "Les Robots d'assistance physique : fonctionnement et modes d'utilisation en industrie," Institut National de Recherche et de Sécurité (INRS), 2017.
- [51] D. Bitonneau, "Conception de systèmes cobotiques industriels : approche robotique avec prise en compte des facteurs humains : application à l'industrie manufacturière au sein de Safran et ArianeGroup," Université de Bordeaux, Français, 2018.
- [52] M. PESHKIN and COLGATE J, "Cobots," *Industrial Robot: An International Journal*, vol. 26, no. 5, pp. 335–341, 1999.
- [53] J.-M. Salotti, E. Ferreri, O. Ly, and D. Daney, "Classification des Systèmes Cobotiques," *Ingénierie cognitive*, vol. 1, no. 1, 2017, doi: 10.21494/ISTE.OP.2018.0268.
- [54] J. E. Colgate, W. Wannasuphprasit, and M. A. Peshkin, "Cobots: Robots for Collaboration With Human Operators," in *Dynamic Systems and Control*, American Society of Mechanical Engineers, Nov. 1996, pp. 433–439. doi: 10.1115/IMECE1996-0367.
- [55] W. Wannasuphprasit, P. Akella, M. Peshkin, and J. E. Colgate, "Cobots : a novel material handling technology," *In International Mechanical Engineering Congress and Exposition*, pp. 1–7, 1998.
- [56] A. Lamour, "Robots collaboratifs : prêts pour le service! Cetim infos," 2016.
-

-
- [57] M. Devy*, “La cobotique : des robots industriels aux robots assistants, coopérants et co-opérateurs,” *Annales des Mines - Réalités industrielles*, 2012(1), pp. 76–85, 2012.
- [58] M. Makarov, “Contribution à la modélisation et la commande robuste de robots manipulateurs à articulations flexibles,” Applications à la robotique interactive, 2013.
- [59] M. Hägele, W. Schaaf, and E. Helms, “Robot Assistants at Manual Workplaces : Effective Co-operation and Safety Aspects,” *In 33rd International Symposium on Robotics*, vol. 7, p. 97, 2022.
- [60] M. Rahimi and W. Karwowski, “A research paradigm in human-robot interaction,” *Int J Ind Ergon*, vol. 5, no. 1, pp. 59–71, Jan. 1990, doi: 10.1016/0169-8141(90)90028-Z.
- [61] T. MOULIÈRES-SEBAN, J. M. SALOTTI, B. CLAVERIE, and D. BITONNEAU, “Classification of cobotic systems for industrial applications,” *proceedings of the 6th workshop towards a Framework for Joint Action*, Oct. 2015.
- [62] C. Bartneck and J. Forlizzi, “A design-centred framework for social human-robot interaction,” in *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759)*, IEEE, pp. 591–594. doi: 10.1109/ROMAN.2004.1374827.
- [63] A. Bicchi, M. A. Peshkin, and J. E. Colgate, “Safety for Physical Human–Robot Interaction,” in *Springer Handbook of Robotics*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1335–1348. doi: 10.1007/978-3-540-30301-5_58.
- [64] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, “Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights,” in *Proceedings of the 13th International Symposium of Robotics Research (ISRR2007)*, Hiroshima, Japan, 2007, pp. 26–29. doi: 10.1007/978-3-642-14743-2_33.
- [65] J. Scholtz, “Theory and evaluation of human robot interactions,” in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, IEEE, 2003, p. 10 pp. doi: 10.1109/HICSS.2003.1174284.
- [66] G. Weiss, “Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence,” in *The MIT press*, London, England , 1999.
- [67] S. J. Russell and P. Norvig, “Artificial Intelligence: A Modern Approach,” in *Pearson Education Limited*, 3rd edition., Malaysia, 2016.
- [68] S. Thrun, W. Burgar, and D. Fox, “Probabilistic Robotics (Intelligent Robotics and Autonomous Agents),” The MIT Press, 2005.
- [69] S. LEMAIGNAN, WARNIER. M, E. SISBOT, and R. ALAMI, “Human-Robot Interaction: Tackling the AI Challenges, Artificial Intelligence,” *Artif Intell*, Jan. 2015.
- [70] T. B. Sheridan, “Human–Robot Interaction: Status and Challenges,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 58, no. 4, pp. 525–532, Jun. 2016, doi: 10.1177/0018720816644364.
- [71] A. Gautam and S. Mohan, “A review of research in multi-robot systems,” in *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, IEEE, Aug. 2012, pp. 1–5. doi: 10.1109/ICIInfS.2012.6304778.
- [72] C. Yifan and S. X. Yang, “A Survey on multi-robot systems,” in *World Automation Congress 2012*, Puerto Vallarta, Mexico: IEEE, Oct. 2012, pp. 1–6.
- [73] T. Arai, E. Pagello, and L. E. Parker, “Guest editorial advances in multirobot systems,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655–661, Oct. 2002, doi: 10.1109/TRA.2002.806024.
-

-
- [74] G. Dudek, Michael R. M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Auton Robots*, vol. 3, no. 4, 1996, doi: 10.1007/BF00240651.
- [75] B. P. Gerkey and M. J. Matarić, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *Int J Rob Res*, vol. 23, no. 9, pp. 939–954, Sep. 2004, doi: 10.1177/0278364904045564.
- [76] C. P. Bechlioulis and K. J. Kyriakopoulos, "Collaborative Multi-Robot Transportation in Obstacle-Cluttered Environments via Implicit Communication," *Front Robot AI*, vol. 5, Aug. 2018, doi: 10.3389/frobt.2018.00090.
- [77] S. F. Everton, "Tracking, destabilizing and disrupting dark networks with social networks analysis," 2008.
- [78] Xinran Wang, D. Green, and J. C. Barca, "Guidelines for improving the robustness of swarm robotic systems through adjustment of network topology," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, Jul. 2017, pp. 1399–1405. doi: 10.1109/AIM.2017.8014214.
- [79] M. Garzón *et al.*, "Using ROS in Multi-robot Systems: Experiences and Lessons Learned from Real-World Field Tests," 2017, pp. 449–483. doi: 10.1007/978-3-319-54927-9_14.
- [80] L. Iocchi, D. Nardi, and M. Salerno, "Reactivity and Deliberation: A Survey on Multi-Robot Systems," 2001, pp. 9–32. doi: 10.1007/3-540-44568-4_2.
- [81] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot Systems: A Classification Focused on Coordination," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2015–2028, Oct. 2004, doi: 10.1109/TSMCB.2004.832155.
- [82] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions," *Auton Robots*, vol. 4, no. 1, pp. 7–27, 1997, doi: 10.1023/A:1008855018923.
- [83] S. A. Deloach, E. T. Matson, and Y. Li, "Exploiting Agent Oriented Software Engineering in Cooperative Robotics Search and Rescue," *Intern J Pattern Recognit Artif Intell*, vol. 17, no. 05, pp. 817–835, Aug. 2003, doi: 10.1142/S0218001403002666.
- [84] S. A. DELOACH, M. F. WOOD, and C. H. SPARKMAN, "MULTIAGENT SYSTEMS ENGINEERING," *International Journal of Software Engineering and Knowledge Engineering*, vol. 11, no. 03, pp. 231–258, Jun. 2001, doi: 10.1142/S0218194001000542.
- [85] L. E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics," in *Distributed Autonomous Robotic Systems 4*, Tokyo: Springer Japan, 2000, pp. 3–12. doi: 10.1007/978-4-431-67919-6_1.
- [86] A. Koubâa and A. Khelil, *Cooperative Robots and Sensor Networks*, vol. 507. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. doi: 10.1007/978-3-642-39301-3.
- [87] B. Adams, C. Huyck, and R. Eguchi, "The Boumerdes (Algeria) Earthquake of May 21, 2003: Preliminary Reconnaissance Using Remotely Sensed Data," , [Online].
- [88] Rédaction VVA, "Casbah d'Alger : guides, tourisme et patrimoine," *Rédaction VVA*, Feb. 2018, [Online]. Available: <https://www.visa-algerie.com/casbah-dalger-guides-tourisme-et-patrimoine/>
- [89] "Essais nucléaires français à Reggane : graves menaces sur la santé et l'environnement," [Online], 2015.
- [90] S. Pirbhulal, H. Zhang, W. Wu, S. C. Mukhopadhyay, and Y.-T. Zhang, "Heartbeats Based Biometric Random Binary Sequences Generation to Secure Wireless Body Sensor Networks," *IEEE Trans Biomed Eng*, vol. 65, no. 12, pp. 2751–2759, Dec. 2018, doi: 10.1109/TBME.2018.2815155.
-

-
- [91] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. M. Leung, "Recent Advances in Industrial Wireless Sensor Networks Toward Efficient Management in IoT," *IEEE Access*, vol. 3, pp. 622–637, 2015, doi: 10.1109/ACCESS.2015.2435000.
- [92] S. Challa *et al.*, "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 69, pp. 534–554, Jul. 2018, doi: 10.1016/j.compeleceng.2017.08.003.
- [93] M. Faheem and V. C. Gungor, "MQRP: Mobile sinks-based QoS-aware data gathering protocol for wireless sensor networks-based smart grid applications in the context of industry 4.0-based on internet of things," *Future Generation Computer Systems*, vol. 82, pp. 358–374, May 2018, doi: 10.1016/j.future.2017.10.009.
- [94] M. M. Hassan, K. Lin, X. Yue, and J. Wan, "A multimedia healthcare data sharing approach through cloud-based body area network," *Future Generation Computer Systems*, vol. 66, pp. 48–58, Jan. 2017, doi: 10.1016/j.future.2015.12.016.
- [95] N. Shakoor, S. Lee, and T. C. Mockler, "High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field," *Curr Opin Plant Biol*, vol. 38, pp. 184–192, Aug. 2017, doi: 10.1016/j.pbi.2017.05.006.
- [96] S. Yang, U. Adeel, Y. Tahir, and J. A. McCann, "Practical Opportunistic Data Collection in Wireless Sensor Networks with Mobile Sinks," *IEEE Trans Mob Comput*, vol. 16, no. 5, pp. 1420–1433, May 2017, doi: 10.1109/TMC.2016.2595574.
- [97] W. Xiong, X. Hu, and T. Jiang, "Measurement and Characterization of Link Quality for IEEE 802.15.4-Compliant Wireless Sensor Networks in Vehicular Communications," *IEEE Trans Industr Inform*, vol. 12, no. 5, pp. 1702–1713, Oct. 2016, doi: 10.1109/TII.2015.2499121.
- [98] S. Lin *et al.*, "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks," *ACM Trans Sens Netw*, vol. 12, no. 1, pp. 1–31, Mar. 2016, doi: 10.1145/2746342.
- [99] N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, New York, NY, USA: ACM, Jul. 2001, pp. 1–14. doi: 10.1145/381677.381679.
- [100] J. Ko and M. Chang, "MoMoRo: Providing Mobility Support for Low-Power Wireless Applications," *IEEE Syst J*, vol. 9, no. 2, pp. 585–594, Jun. 2015, doi: 10.1109/JSYST.2014.2299592.
- [101] Michael A Leabman and Gregory Scott Brewer., "Receivers for wireless power transmission. ," *US Patent 9,912,199*, Mar. 2018.
- [102] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Adaptive and Channel-Aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks," *IEEE Trans Wirel Commun*, vol. 15, no. 5, pp. 3718–3731, May 2016, doi: 10.1109/TWC.2016.2526601.
- [103] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Trans Sens Netw*, vol. 6, no. 2, pp. 1–49, Feb. 2010, doi: 10.1145/1689239.1689246.
- [104] Chipcon, "2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver," *Reference SWRS041c (2013)*, p. 96, 2004.
- [105] LAN/MAN Standards Committee, "IEEE Standard for information technology telecommunications and information exchange betweensystems local and metropolitan area networks specific requirements. Part 15.4: wireless Medium Access Control (MAC) and Physical Layer (PHY)specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Computer Society (IEEE Std 802)*, Jan. 2007.
-

-
- [106] R. D. Gomes, D. V. Queiroz, A. C. Lima Filho, I. E. Fonseca, and M. S. Alencar, “Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes,” *Ad Hoc Networks*, vol. 59, pp. 116–133, May 2017, doi: 10.1016/j.adhoc.2017.02.007.
- [107] D. Yuan, S. S. Kanhere, and M. Hollick, “Instrumenting Wireless Sensor Networks — A survey on the metrics that matter,” *Pervasive Mob Comput*, vol. 37, pp. 45–62, Jun. 2017, doi: 10.1016/j.pmcj.2016.10.001.
- [108] N. Baccour *et al.*, “Radio link quality estimation in wireless sensor networks,” *ACM Trans Sens Netw*, vol. 8, no. 4, pp. 1–33, Sep. 2012, doi: 10.1145/2240116.2240123.
- [109] E. Miluzzo, X. Zheng, K. Fodor, and A. T. Campbell, “Radio Characterization of 802.15.4 and Its Impact on the Design of Mobile Sensor Networks,” in *Wireless Sensor Networks*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 171–188. doi: 10.1007/978-3-540-77690-1_11.
- [110] N. Ahmed, S. S. Kanhere, and S. Jha, “On the importance of link characterization for aerial wireless sensor networks,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 52–57, May 2016, doi: 10.1109/MCOM.2016.7470935.
- [111] J. Wen and W. Dargie, “Characterization of Link Quality Fluctuation in Mobile Wireless Sensor Networks,” *ACM Transactions on Cyber-Physical Systems*, vol. 5, no. 3, pp. 1–24, Jul. 2021, doi: 10.1145/3448737.
- [112] Qian Dong and W. Dargie, “Evaluation of the reliability of RSSI for indoor localization,” in *2012 International Conference on Wireless Communications in Underground and Confined Areas*, IEEE, Aug. 2012, pp. 1–6. doi: 10.1109/ICWCUCA.2012.6402492.
- [113] H. Nikookar and H. Hashemi, “Statistical modeling of signal amplitude fading of indoor radio propagation channels,” in *Proceedings of 2nd IEEE International Conference on Universal Personal Communications*, IEEE, 1993, pp. 84–88. doi: 10.1109/ICUPC.1993.528353.
- [114] W. Su, S.-J. Lee, and M. Gerla, “Mobility prediction in wireless networks,” in *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No.00CH37155)*, IEEE, 1999, pp. 491–495. doi: 10.1109/MILCOM.2000.905001.
- [115] F. De Rosa, A. Malizia, and M. Mecella, “Disconnection Prediction in Mobile Ad hoc Networks for Supporting Cooperative Work,” *IEEE Pervasive Comput*, vol. 4, no. 3, pp. 62–70, Jul. 2005, doi: 10.1109/MPRV.2005.55.
- [116] S. Chen, H. Jones, and D. Jayalath, “Effective link operation duration: a new routing metric for mobile ad hoc networks,” in *Proceedings of the 1st International Conference on Signal Processing and Communication Systems 2007, incorporating the 9th International Symposium on DSP and Communication Systems and the 6th Workshop on the Internet, Telecommunications and Signal Pr.* pp. 315–320, 2008. [Online]. Available: <http://eprints.qut.edu.au/9913/1/ChenJonesJayalathISCPSC07.pdf>
- [117] A. J. Nicholson and B. D. Noble, “BreadCrumbs,” in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, New York, NY, USA: ACM, Sep. 2008, pp. 46–57. doi: 10.1145/1409944.1409952.
- [118] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. K. Das, “Link quality prediction in mesh networks,” *Comput Commun*, vol. 31, no. 8, pp. 1497–1512, May 2008, doi: 10.1016/j.comcom.2008.01.047.
- [119] R. Sabitha and T. Thyagarajan, “Fuzzy logic-based transmission power control algorithm for energy efficient MAC protocol in wireless sensor networks,” *International Journal of Communication Networks and Distributed Systems*, vol. 9, no. 3/4, p. 247, 2012, doi: 10.1504/IJCNDS.2012.048873.
-

-
- [120] R. K. Guha and S. Sarkar, "Characterizing temporal SNR variation in 802.11 networks," in *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, IEEE, 2006, pp. 1408–1413. doi: 10.1109/WCNC.2006.1696493.
- [121] G. M. de Araujo, J. Kaiser, and L. B. Becker, "An optimized Markov model to predict link quality in mobile wireless sensor networks," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, Jul. 2012, pp. 000307–000312. doi: 10.1109/ISCC.2012.6249313.
- [122] A. Chella, G. Lo Re, I. Macaluso, M. Ortolani, and D. Peri, "Multi-robot Interacting Through Wireless Sensor Networks," in *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, vol. 4733 LNAI, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 789–796. doi: 10.1007/978-3-540-74782-6_70.
- [123] A. Erman, L. Hoesel, P. Havinga, and J. Wu, "Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management," *IEEE Wirel Commun*, vol. 15, no. 6, pp. 38–46, Dec. 2008, doi: 10.1109/MWC.2008.4749746.
- [124] E. P. De Freitas *et al.*, "Coordinating Aerial Robots and Unattended Ground Sensors for Intelligent Surveillance Systems," *International Journal of Computers Communications & Control*, vol. 5, no. 1, p. 52, Mar. 2010, doi: 10.15837/ijccc.2010.1.2464.
- [125] J. Valente, D. Sanz, A. Barrientos, J. del Cerro, Á. Ribeiro, and C. Rossi, "An Air-Ground Wireless Sensor Network for Crop Monitoring," *Sensors*, vol. 11, no. 6, pp. 6088–6108, Jun. 2011, doi: 10.3390/s110606088.
- [126] Z. Beck, W. T. L. Teacy, A. Rogers, and N. R. Jennings, "Collaborative online planning for automated victim search in disaster response," *Rob Auton Syst*, vol. 100, pp. 251–266, Feb. 2018, doi: 10.1016/j.robot.2017.09.014.
- [127] A. Ali, L. A. Latiff, and N. Fisal, "GPS-free indoor location tracking in mobile ad hoc network (MANET) using RSSI," in *2004 RF and Microwave Conference (IEEE Cat. No.04EX924)*, IEEE, 2005, pp. 251–255. doi: 10.1109/RFM.2004.1411119.
- [128] G. C. K. & C. J. Deak, "Filters for RSSI-based measurements in a device-free passive localisation scenario," *Image Processing & Communications*, vol. 15, no. 1, pp. 23–34, 2010.
- [129] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. 1989.
- [130] M. Armada and P. Gonzalez de Santos, "Climbing, walking and intervention robots," *Industrial Robot: An International Journal*, vol. 24, no. 2, pp. 158–163, Apr. 1997, doi: 10.1108/01439919710165716.
- [131] P. Wilke and T. Bräunl, "Flexible wireless communication network for mobile robot agents," *Industrial Robot: An International Journal*, vol. 28, no. 3, pp. 220–232, Jun. 2001, doi: 10.1108/01439910110389380.
- [132] Q. V. Khanh, N. V. Hoai, L. D. Manh, A. N. Le, and G. Jeon, "Wireless Communication Technologies for IoT in 5G: Vision, Applications, and Challenges," *Wirel Commun Mob Comput*, vol. 2022, pp. 1–12, Feb. 2022, doi: 10.1155/2022/3229294.
- [133] M. Sikimic, M. Amovic, V. Vujovic, B. Suknovic, and D. Manjak, "An Overview of Wireless Technologies for IoT Network," in *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, IEEE, Mar. 2020, pp. 1–6. doi: 10.1109/INFOTEH48170.2020.9066337.
- [134] D. Bapna, E. Rollins, A. Foessel, and R. Whittaker, "Antenna pointing for high bandwidth communications from mobile robots," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, IEEE, 1998, pp. 3468–3473. doi: 10.1109/ROBOT.1998.680974.
- [135] J. V. S. Teixeira, A. M. Reis, F. B. Mendes, and L. G. L. Vergara, "Collaborative Robots," in *Studies in Systems, Decision and Control*, vol. 202, 2019, pp. 791–796. doi: 10.1007/978-3-030-14730-3_83.
-

- [136] A. Seferagić, J. Famaey, E. De Poorter, and J. Hoebeke, “Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things,” *Sensors*, vol. 20, no. 2, p. 488, Jan. 2020, doi: 10.3390/s20020488.
- [137] P. Indira, I. S. Arafat, R. Karthikeyan, S. Selvarajan, and P. K. Balachandran, “Fabrication and investigation of agricultural monitoring system with IoT & AI,” *SN Appl Sci*, vol. 5, no. 12, p. 322, Dec. 2023, doi: 10.1007/s42452-023-05526-1.
- [138] C. Shim *et al.*, “Development of a Work Environment Monitoring System for Improving HSE and Production Information Management Within a Shipyard Based on Wireless Communication,” *Journal of the Society of Naval Architects of Korea*, vol. 60, no. 5, pp. 367–374, 2023, doi: 10.3744/snak.2023.60.5.367.
- [139] A. del Bosque, “Tesis doctoral: Desarrollo de sensores flexibles piezorresistivos basados en polímeros dopados con nanoestructuras de carbono para aplicaciones biomédicas,” 2023. [Online]. Available: <https://dialnet.unirioja.es/servlet/tesis?codigo=323203>
- [140] J. Yao, H. Huang, J. Su, R. Xie, X. Zheng, and K. Wu, “Enabling Cross-Technology Coexistence for ZigBee Devices through Payload Encoding,” *IEEE Trans Mob Comput*, pp. 1–18, 2023, doi: 10.1109/TMC.2023.3345830.
- [141] B. V. Krishnaveni, A. Venkatalakshmi, and B. R. Kumar, “Localization for IoT Based Systems Using UWB,” in *Contemporary Perspective on Science, Technology and Research Vol. 2*, B P International, 2023, pp. 12–26. doi: 10.9734/bpi/cpstr/v2/11383F.
- [142] R. Mukhia, K. G. Sarambage Jayarathna, and A. Lertsinsrubtavee, “Performance Evaluation of LoRaWAN Forest Fire Monitoring Network in the Wild,” in *Proceedings of the 18th Asian Internet Engineering Conference*, New York, NY, USA: ACM, Dec. 2023, pp. 96–104. doi: 10.1145/3630590.3630602.
- [143] M. A. Al-Shareeda, A. Abdullah Alsadhan, H. H. Qasim, and S. Manickam, “Long range technology for internet of things: review, challenges, and future directions,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 6, pp. 3758–3767, Dec. 2023, doi: 10.11591/eei.v12i6.5214.
- [144] S. A. Magsi, M. H. B. M. Khir, I. B. M. Nawi, A. Saboor, and M. A. Siddiqui, “Support Vector Regression based Localization Approach using LoRaWAN,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, pp. 307–312, 2023, doi: 10.14569/IJACSA.2023.0140335.
- [145] H. F. FakhruLdeen, M. J. Saadh, S. Khan, N. A. Salim, N. Jhamat, and G. Mustafa, “Enhancing smart home device identification in WiFi environments for futuristic smart networks-based IoT,” *Int J Data Sci Anal*, Jan. 2024, doi: 10.1007/s41060-023-00489-3.
- [146] X. Xie, D. Zhang, Y. Li, Y. Hu, Q. Sun, and Y. Chen, “Robust WiFi Respiration Sensing in the Presence of Interfering Individual,” *IEEE Trans Mob Comput*, pp. 1–17, 2024, doi: 10.1109/TMC.2023.3348879.
- [147] B. Sheng, R. Han, H. Cai, F. Xiao, L. Gui, and Z. Guo, “CDFi: Cross-Domain Action Recognition using WiFi Signals,” *IEEE Trans Mob Comput*, pp. 1–16, 2024, doi: 10.1109/TMC.2023.3348939.
- [148] Z. Wang, Y. Liu, and Y. Dong, “Novel Miniaturized Circularly Polarized Inverted-F Antenna with Planar Configuration,” *IEEE Antennas Wirel Propag Lett*, pp. 1–5, 2023, doi: 10.1109/LAWP.2023.3341843.
- [149] A. Akasaka, F. Lu, A. Yamaguchi, K. Takeuchi, and H. Shinbo, “Measurements of Reflection Characteristics of the Lunar Surface for Radio Propagation,” in *2023 IEEE Aerospace Conference*, IEEE, Mar. 2023, pp. 1–10. doi: 10.1109/AERO55745.2023.10115879.
- [150] P. Y. Chan, J.-C. Chao, and R.-B. Wu, “A Wi-Fi-Based Passive Indoor Positioning System via Entropy-Enhanced Deployment of Wi-Fi Sniffers,” *Sensors*, vol. 23, no. 3, p. 1376, Jan. 2023, doi: 10.3390/s23031376.

-
- [151] E. Y. Cho, J. U. Kwon, M. S. Chae, S. Y. Cho, J. Yoo, and S. Seo, "Indoor Positioning Technology Integrating Pedestrian Dead Reckoning and WiFi Fingerprinting Based on EKF with Adaptive Error Covariance," vol. 12, no. 3, pp. 271–280, 2023, doi: 10.11003/JPNT.2023.12.3.27.
- [152] Y. S. Handayani and Y. Mardiana, "KENDALI ROBOT BLUETOOTH DENGAN SMARTPHONE ANDROID BERBASIS ARDUINO UNO," *ILKOM Jurnal Ilmiah*, vol. 10, no. 3, pp. 331–337, Dec. 2018, doi: 10.33096/ilkom.v10i3.363.331-337.
- [153] R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A. Ijspeert, "YaMoR and Bluemove — An Autonomous Modular Robot with Bluetooth Interface for Exploring Adaptive Locomotion," in *Climbing and Walking Robots*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 685–692. doi: 10.1007/3-540-26415-9_82.
- [154] Yeong Che Fai, S. H. M. Amin, N. bt Fisal, and J. A. Bakar, "Bluetooth enabled mobile robot," in *2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02.*, IEEE, 2002, pp. 903–908. doi: 10.1109/ICIT.2002.1189288.
- [155] R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A. Jan Ijspeert, "Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface," *Industrial Robot: An International Journal*, vol. 33, no. 4, pp. 285–290, Jul. 2006, doi: 10.1108/01439910610667908.
- [156] A. Murad, O. Bayat, and H. M. Marhoon, "Implementation of rover tank firefighting robot for closed areas based on arduino microcontroller," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, p. 56, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp56-63.
- [157] M. Xiao, Q. Wen, L. Ji, and W. Wang, "Tennis Ball Picking Robot Based on Bluetooth Control," in *2021 2nd International Conference on Control, Robotics and Intelligent System*, New York, NY, USA: ACM, Aug. 2021, pp. 12–16. doi: 10.1145/3483845.3483848.
- [158] K. Weinmann and S. Simske, "Design of Bluetooth 5.1 Angle of Arrival Homing Controller for Autonomous Mobile Robot," *Robotics*, vol. 12, no. 4, p. 115, Aug. 2023, doi: 10.3390/robotics12040115.
- [159] A. Fernandes, M. S. Couceiro, D. Portugal, J. Machado Santos, and R. P. Rocha, "Ad hoc communication in teams of mobile robots using zigbee technology," *Computer Applications in Engineering Education*, vol. 23, no. 5, pp. 733–745, Sep. 2015, doi: 10.1002/cae.21646.
- [160] J. Wan, Y. Wang, Q. Qin, and Y. Li, "Multi-robots' communication system based on ZigBee network," in *2009 9th International Conference on Electronic Measurement & Instruments*, IEEE, Aug. 2009, pp. 3-14-3–19. doi: 10.1109/ICEMI.2009.5274350.
- [161] I. M. A. Nugraha, I. G. M. N. Desnanjaya, I. W. D. Pranata, and W. Harianto, "Stability Data Xbee S2b Zigbee Communication on Arduino Based Sumo Robot," *Journal of Robotics and Control (JRC)*, vol. 2, no. 3, pp. 153–160, 2021, doi: 10.18196/jrc.2370.
- [162] W. Ying *et al.*, "Research on Key Technologies of Intelligent Seeding Robot Based on Zigbee Network," *IOP Conf Ser Mater Sci Eng*, vol. 631, no. 3, p. 032042, Oct. 2019, doi: 10.1088/1757-899X/631/3/032042.
- [163] A. Sh. M. Al-Obaidi, A. Al-Qassar, A. R. Nasser, A. Alkhayyat, A. J. Humaidi, and I. K. Ibraheem, "Embedded Design and Implementation of Mobile Robot for Surveillance Applications," *Indonesian Journal of Science and Technology*, vol. 6, no. 2, pp. 427–440, May 2021, doi: 10.17509/ijost.v6i2.36275.
- [164] S. Kanaga Suba Raja, V. Balaji, and M. Vivekanandan, "Smart robot for disaster detection using zigbee technology," *International Journal of Applied Engineering Research*, vol. 10, no. 10, pp. 27311–27320, 2015.
-

-
- [165] R. Karothia and M. K. Chattopadhyay, "Review of Various Technologies Involved in Precision Farming Automation," in *Precision Agriculture for Sustainability*, New York: Apple Academic Press, 2023, pp. 3–20. doi: 10.1201/9781003435228-2.
- [166] K. Z. Islam, D. Murray, D. Diepeveen, M. G. K. Jones, and F. Sohel, "LoRa-based outdoor localization and tracking using unsupervised symbolization," *Internet of Things*, vol. 25, p. 101016, Apr. 2024, doi: 10.1016/j.iot.2023.101016.
- [167] B. Paul, C. Assi, G. Kaddoum, R. Palit, and S. Islam, "An Energy-Efficient Cluster Formation Based On Optimal Node Distribution in Full Capacity Multi-Hop LoRa Networks," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1–1, 2024, doi: 10.1109/OJCOMS.2024.3350511.
- [168] M. P. Manuel and K. Daimi, "Implementing cryptography in LoRa based communication devices for unmanned ground vehicle applications," *SN Appl Sci*, vol. 3, no. 4, p. 397, Apr. 2021, doi: 10.1007/s42452-021-04377-y.
- [169] T. Maneekittichote and T. Chanthasopeephan, "Mobile Robot Swarm Navigation and Communication Using LoRaWan," in *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, IEEE, Feb. 2020, pp. 22–25. doi: 10.1109/ICMRE49073.2020.9064973.
- [170] F. Ndimumahoro, M. S. Stepanov, A. R. Muzata, D. Tammam, and V. G. Popov, "Using the Principles of Mobile Systems Modeling for LoRaWAN Characteristics Estimation," in *2022 Systems of Signals Generating and Processing in the Field of on Board Communications*, IEEE, Mar. 2022, pp. 1–8. doi: 10.1109/IEEECONF53456.2022.9744363.
- [171] R. Dai, O. Diraneyya, and S. Brell-Çokcan, "Improving data communication on construction sites via LoRaWAN," *Construction Robotics*, vol. 5, no. 2, pp. 87–100, Jun. 2021, doi: 10.1007/s41693-021-00059-8.
- [172] M. A. M. Almuhaaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions," *Electronics (Basel)*, vol. 11, no. 1, p. 164, Jan. 2022, doi: 10.3390/electronics11010164.
- [173] M. Trnka, J. Svacina, T. Cerny, E. Song, J. Hong, and M. Bures, "Securing Internet of Things Devices Using The Network Context," *IEEE Trans Industr Inform*, vol. 16, no. 6, pp. 4017–4027, Jun. 2020, doi: 10.1109/TII.2019.2954100.
- [174] M. Khari, A. K. Garg, A. H. Gandomi, R. Gupta, R. Patan, and B. Balusamy, "Securing Data in Internet of Things (IoT) Using Cryptography and Steganography Techniques," *IEEE Trans Syst Man Cybern Syst*, vol. 50, no. 1, pp. 73–80, Jan. 2020, doi: 10.1109/TSMC.2019.2903785.
- [175] T. Abderrahmane, N. Amardjia, and T. Mohammed, "Securing laboratories through internet of things networks: an ocomprehensive approach for ensuring safety and efficiency," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 572–585, Feb. 2024, doi: 10.11591/eei.v13i1.6728.
- [176] A. Singh Rajawat *et al.*, "Reformist Framework for Improving Human Security for Mobile Robots in Industry 4.0," *Mobile Information Systems*, vol. 2021, pp. 1–10, Oct. 2021, doi: 10.1155/2021/4744220.
- [177] A. Khalid, P. Kirisci, Z. H. Khan, Z. Ghairi, K.-D. Thoben, and J. Pannek, "Security framework for industrial collaborative robotic cyber-physical systems," *Comput Ind*, vol. 97, pp. 132–145, May 2018, doi: 10.1016/j.compind.2018.02.009.
- [178] S. Chakrabarty and D. W. Engels, "A secure IoT architecture for Smart Cities," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, Jan. 2016, pp. 812–813. doi: 10.1109/CCNC.2016.7444889.
-

-
- [179] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the Internet of Things (IoT): A Security Taxonomy for IoT," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, Aug. 2018, pp. 163–168. doi: 10.1109/TrustCom/BigDataSE.2018.00034.
- [180] M. Horton, Lei Chen, and B. Samanta, "Enhancing the security of IoT enabled robotics: Protecting TurtleBot file system and communication," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, Jan. 2017, pp. 662–666. doi: 10.1109/ICCNC.2017.7876208.
- [181] S. Haas, T. Ulz, and C. Steger, "Secured action authorization for industrial mobile robots," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, IEEE, May 2018, pp. 806–811. doi: 10.1109/ICPHYS.2018.8390810.
- [182] J.-P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations," *Int J Inf Secur*, vol. 21, no. 1, pp. 115–158, Feb. 2022, doi: 10.1007/s10207-021-00545-8.
- [183] M. Mukhandi, D. Portugal, S. Pereira, and M. S. Couceiro, "A novel solution for securing robot communications based on the MQTT protocol and ROS," in *2019 IEEE/SICE International Symposium on System Integration (SII)*, IEEE, Jan. 2019, pp. 608–613. doi: 10.1109/SII.2019.8700390.
- [184] J. Soldatos, S. Kyriazakos, P. Ziafati, and A. Mihovska, "Securing IoT Applications with Smart Objects: Framework and a Socially Assistive Robots Case Study," *Wirel Pers Commun*, vol. 117, no. 1, pp. 261–280, Mar. 2021, doi: 10.1007/s11277-020-07039-1.
- [185] F. J. Fernández-Jiménez and J. R. M. Dios, "A Robot–Sensor Network Security Architecture for Monitoring Applications," *IEEE Internet Things J*, vol. 9, no. 8, pp. 6288–6304, Apr. 2022, doi: 10.1109/JIOT.2021.3113756.
- [186] M. Alramadhan and K. Sha, "An Overview of Access Control Mechanisms for Internet of Things," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, Jul. 2017, pp. 1–6. doi: 10.1109/ICCCN.2017.8038503.
- [187] K. Ahmad Yousef, A. AlMajali, S. Ghalyon, W. Dweik, and B. Mohd, "Analyzing Cyber-Physical Threats on Robotic Platforms," *Sensors*, vol. 18, no. 5, p. 1643, May 2018, doi: 10.3390/s18051643.
- [188] A. Vulpe, R. Crăciunescu, A.-M. Drăgulescu, S. Kyriazakos, A. Paikan, and P. Ziafati, "Enabling Security Services in Socially Assistive Robot Scenarios for Healthcare Applications," *Sensors*, vol. 21, no. 20, p. 6912, Oct. 2021, doi: 10.3390/s21206912.
- [189] B. Al-Duwairi, W. Al-Kahla, M. A. AlRefai, Y. Abedalqader, A. Rawash, and R. Fahmawi, "SIEM-based detection and mitigation of IoT-botnet DDoS attacks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, p. 2182, Apr. 2020, doi: 10.11591/ijece.v10i2.pp2182-2191.
- [190] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2710–2723, Nov. 2013, doi: 10.1016/j.adhoc.2013.05.003.
- [191] R. Podder and R. K. Barai, "Hybrid Encryption Algorithm for the Data Security of ESP32 based IoT-enabled Robots," in *2021 Innovations in Energy Management and Renewable Resources(52042)*, IEEE, Feb. 2021, pp. 1–5. doi: 10.1109/IEMRE52042.2021.9386824.
- [192] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An Experimental Security Analysis of an Industrial Robot Controller," in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, May 2017, pp. 268–286. doi: 10.1109/SP.2017.20.
-

-
- [193] P. P. Ray, "Internet of Robotic Things: Concept, Technologies, and Challenges," *IEEE Access*, vol. 4, pp. 9489–9500, 2016, doi: 10.1109/ACCESS.2017.2647747.
- [194] B. Sudharsan *et al.*, "OTA-TinyML: Over the Air Deployment of TinyML Models and Execution on IoT Devices," *IEEE Internet Comput*, vol. 26, no. 3, pp. 69–78, May 2022, doi: 10.1109/MIC.2021.3133552.
- [195] M. Wazid, A. K. Das, and J.-H. Lee, "User authentication in a tactile internet based remote surgery environment: Security issues, challenges, and future research directions," *Pervasive Mob Comput*, vol. 54, pp. 71–85, Mar. 2019, doi: 10.1016/j.pmcj.2019.02.004.
- [196] A. T. Abu-Jassar, H. Attar, V. Lyashenko, A. Amer, S. Sotnik, and A. Solyman, "Access Control to Robotic Systems Based on Biometric: The Generalized Model and its Practical Implementation," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 5, pp. 313–328, Oct. 2023, doi: 10.22266/ijies2023.1031.27.
- [197] S. Jain and R. Doriya, "Authentication of Robots Using ECC to Access Cloud-Based Services," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 57, 2021, pp. 861–870. doi: 10.1007/978-981-15-9509-7_69.
- [198] A. A. El Kalam, A. Outchakoucht, and H. Es-Samaali, "Emergence-Based Access Control: New Approach to Secure the Internet of Things," in *Proceedings of the 1st International Conference on Digital Tools & Uses Congress - DTUC '18*, New York, New York, USA: ACM Press, 2018, pp. 1–11. doi: 10.1145/3240117.3240136.
- [199] A. Ometov, V. Petrov, S. Bezzateev, S. Andreev, Y. Koucheryavy, and M. Gerla, "Challenges of Multi-Factor Authentication for Securing Advanced IoT Applications," *IEEE Netw*, vol. 33, no. 2, pp. 82–88, Mar. 2019, doi: 10.1109/MNET.2019.1800240.
- [200] A. A. Althahir *et al.*, "Optimizing Visual Sensor Coverage Overlaps for Multiview Surveillance Systems," *IEEE Sens J*, vol. 18, no. 11, pp. 4544–4552, Jun. 2018, doi: 10.1109/JSEN.2018.2825781.
- [201] Zhijun Tang and U. Ozguner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 898–908, Oct. 2005, doi: 10.1109/TRO.2005.847567.
- [202] C. K. H. Lee, G. T. S. Ho, K. L. Choy, and G. K. H. Pang, "A RFID-based recursive process mining system for quality assurance in the garment industry," *Int J Prod Res*, vol. 52, no. 14, pp. 4216–4238, Jul. 2014, doi: 10.1080/00207543.2013.869632.
- [203] M. & B. M. Hefeeda, "Forest fire modeling and early detection using wireless sensor networks," *Ad Hoc Sens. Wirel. Networks*, vol. 7, no. 3–4, pp. 169–224, 2009.
- [204] J. Zhang, W. Li, N. Han, and J. Kan, "Forest fire detection system based on a ZigBee wireless sensor network," *Frontiers of Forestry in China*, vol. 3, no. 3, pp. 369–374, Sep. 2008, doi: 10.1007/s11461-008-0054-3.
- [205] I. Tkach and Y. Edan, *Distributed Heterogeneous Multi Sensor Task Allocation Systems*, vol. 7. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-34735-2.
- [206] F. Ducatelle, G. A. Di Caro, C. Pinciroli, F. Mondada, and L. M. Gambardella, "Communication assisted navigation in robotic swarms: Self-organization and cooperation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Sep. 2011, pp. 4981–4988. doi: 10.1109/IROS.2011.6094454.
- [207] Á. Gutiérrez, A. Campo, F. Monasterio-Huelin, L. Magdalena, and M. Dorigo, "Collective decision-making based on social odometry," *Neural Comput Appl*, vol. 19, no. 6, pp. 807–823, Sep. 2010, doi: 10.1007/s00521-010-0380-x.
-

-
- [208] A. Jevtic, Á. Gutierrez, D. Andina, and M. Jamshidi, “Distributed Bees Algorithm for Task Allocation in Swarm of Robots,” *IEEE Syst J*, vol. 6, no. 2, pp. 296–304, Jun. 2012, doi: 10.1109/JSYST.2011.2167820.
- [209] B. B. Werger and M. J. Matarić, “Broadcast of Local Eligibility for Multi-Target Observation,” in *Distributed Autonomous Robotic Systems 4*, Tokyo: Springer Japan, 2000, pp. 347–356. doi: 10.1007/978-4-431-67919-6_33.
- [210] I. Tkach, A. Jevtić, S. Nof, and Y. Edan, “A Modified Distributed Bees Algorithm for Multi-Sensor Task Allocation,” *Sensors*, vol. 18, no. 3, p. 759, Mar. 2018, doi: 10.3390/s18030759.
- [211] L. Liu, N. Michael, and D. A. Shell, “Communication constrained task allocation with optimized local task swaps,” *Auton Robots*, vol. 39, no. 3, pp. 429–444, Oct. 2015, doi: 10.1007/s10514-015-9481-9.
- [212] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-Based Multirobot Coordination: A Survey and Analysis,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, Jul. 2006, doi: 10.1109/JPROC.2006.876939.
- [213] C. Robin and S. Lacroix, “Multi-robot target detection and tracking: taxonomy and survey,” *Auton Robots*, vol. 40, no. 4, pp. 729–760, Apr. 2016, doi: 10.1007/s10514-015-9491-7.
- [214] B. L. Brumitt and A. Stentz, “GRAMMPS: a generalized mission planner for multiple mobile robots in unstructured environments,” in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, IEEE, 1998, pp. 1564–1571. doi: 10.1109/ROBOT.1998.677360.
- [215] P. Caloud, Wonyun Choi, J.-C. Latombe, C. Le Pape, and M. Yim, “Indoor automation with many mobile robots,” in *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, IEEE, 1990, pp. 67–72. doi: 10.1109/IROS.1990.262370.
- [216] M. G. Ball, B. Qela, and S. Wesolkowski, “A Review of the Use of Computational Intelligence in the Design of Military Surveillance Networks,” 2016, pp. 663–693. doi: 10.1007/978-3-319-26450-9_24.
- [217] A. Khamis, A. Hussein, and A. Elmogy, “Multi-robot Task Allocation: A Review of the State-of-the-Art,” 2015, pp. 31–51. doi: 10.1007/978-3-319-18299-5_2.
- [218] H. Rowaihy *et al.*, “A survey of sensor selection schemes in wireless sensor networks,” E. M. Carapezza, Ed., Apr. 2007, p. 65621A. doi: 10.1117/12.723514.
- [219] L. Bayindir, “A review of swarm robotics tasks,” *Neurocomputing*, vol. 172, pp. 292–321, Jan. 2016, doi: 10.1016/j.neucom.2015.05.116.
- [220] P. Li and H. Duan, “Bio-inspired Computation Algorithms,” in *Bio-inspired Computation in Unmanned Aerial Vehicles*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 35–69. doi: 10.1007/978-3-642-41196-0_2.
- [221] A. Jevtić, “Swarm intelligence: novel tools for optimization, feature extraction, and multi-agent system modeling,” Universidad Politécnica de Madrid, 2011. doi: 10.20868/UPM.thesis.7206.
- [222] C. Blum and R. Groß, “Swarm Intelligence in Optimization and Robotics,” in *Springer Handbook of Computational Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 1291–1309. doi: 10.1007/978-3-662-43505-2_66.
- [223] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999. doi: 10.1093/oso/9780195131581.001.0001.
- [224] Z. Sun, Y. Liu, and L. Tao, “Attack localization task allocation in wireless sensor networks based on multi-objective binary particle swarm optimization,” *Journal of Network and Computer Applications*, vol. 112, pp. 29–40, Jun. 2018, doi: 10.1016/j.jnca.2018.03.023.
-

-
- [225] R. Groß, S. Nouyan, M. Bonani, F. Mondada, and M. Dorigo, “Division of Labour in Self-organised Groups,” in *From Animals to Animats 10*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 426–436. doi: 10.1007/978-3-540-69134-1_42.
- [226] D. Barbagallo, E. Di Nitto, D. J. Dubois, and R. Mirandola, “A Bio-inspired Algorithm for Energy Optimization in a Self-organizing Data Center,” 2010, pp. 127–151. doi: 10.1007/978-3-642-14412-7_7.
- [227] D. & B. P. Upadhyay, “An energy efficient proposed framework for time synchronization problem of wireless sensor network,” In *Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA*. Springer India , vol. 3, pp. 377–385, 2016.
- [228] S. Berman, A. Halasz, V. Kumar, and S. Pratt, “Bio-Inspired Group Behaviors for the Deployment of a Swarm of Robots to Multiple Destinations,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, Apr. 2007, pp. 2318–2323. doi: 10.1109/ROBOT.2007.363665.
- [229] Y. Tan and K. Ding, “A Survey on GPU-Based Implementation of Swarm Intelligence Algorithms,” *IEEE Trans Cybern*, vol. 46, no. 9, pp. 2028–2041, Sep. 2016, doi: 10.1109/TCYB.2015.2460261.
- [230] A. Jevtić, D. Andina, and M. Jamshidi, “Distributed Task Allocation in Swarms of Robots,” in *Robotics*, IGI Global, 2014, pp. 450–473. doi: 10.4018/978-1-4666-4607-0.ch023.
- [231] T. H. Labella, M. Dorigo, and J.-L. Deneubourg, “Division of labor in a group of robots inspired by ants’ foraging behavior,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 1, no. 1, pp. 4–25, Sep. 2006, doi: 10.1145/1152934.1152936.
- [232] S. Z. Zhang and C. K. M. Lee, “An Improved Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, Oct. 2015, pp. 2124–2128. doi: 10.1109/SMC.2015.371.
- [233] B. Xue, M. Zhang, and W. N. Browne, “Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach,” *IEEE Trans Cybern*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013, doi: 10.1109/TSMCB.2012.2227469.
- [234] B. Fu, Y. Xiao, X. Liang, and C. L. Philip Chen, “Bio-inspired Group Modeling and Analysis for Intruder Detection in Mobile Sensor/Robotic Networks,” *IEEE Trans Cybern*, vol. 45, no. 1, pp. 103–115, Jan. 2015, doi: 10.1109/TCYB.2014.2320717.
- [235] N. Quijano and K. M. Passino, “Honey bee social foraging algorithms for resource allocation: Theory and application,” *Eng Appl Artif Intell*, vol. 23, no. 6, pp. 845–861, Sep. 2010, doi: 10.1016/j.engappai.2010.05.004.
- [236] M. Senanayake, I. Senthoran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, “Search and tracking algorithms for swarms of robots: A survey,” *Rob Auton Syst*, vol. 75, pp. 422–434, Jan. 2016, doi: 10.1016/j.robot.2015.08.010.
- [237] J. Cao, B. Yin, X. Lu, Y. Kang, and X. Chen, “A modified artificial bee colony approach for the 0-1 knapsack problem,” *Applied Intelligence*, vol. 48, no. 6, pp. 1582–1595, Jun. 2018, doi: 10.1007/s10489-017-1025-x.
- [238] Eberhart RC, Shi Y, and Kennedy J, *Swarm Intelligence*. Elsevier, 2001. doi: 10.1016/B978-1-55860-595-4.X5000-1.
- [239] A. Hussein, P. Marín-Plaza, F. García, and J. M. Armingol, “Hybrid Optimization-Based Approach for Multiple Intelligent Vehicles Requests Allocation,” *J Adv Transp*, vol. 2018, pp. 1–11, 2018, doi: 10.1155/2018/2493401.
-

-
- [240] N. Kalra, D. Ferguson, and A. Stentz, “Hoplites: A Market-Based Framework for Planned Tight Coordination in Multirobot Teams,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 1170–1177. doi: 10.1109/ROBOT.2005.1570274.
- [241] R. Balamurugan, A. M. Natarajan, and K. Premalatha, “Stellar-Mass Black Hole Optimization for Biclustering Microarray Gene Expression Data,” *Applied Artificial Intelligence*, vol. 29, no. 4, pp. 353–381, Apr. 2015, doi: 10.1080/08839514.2015.1016391.
- [242] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, “A survey on metaheuristics for stochastic combinatorial optimization,” *Nat Comput*, vol. 8, no. 2, pp. 239–287, Jun. 2009, doi: 10.1007/s11047-008-9098-4.
- [243] K. Sörensen, M. Sevaux, and F. Glover, “A History of Metaheuristics,” in *Handbook of Heuristics*, Cham: Springer International Publishing, 2018, pp. 791–808. doi: 10.1007/978-3-319-07124-4_4.
- [244] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization,” *ACM Comput Surv*, vol. 35, no. 3, pp. 268–308, Sep. 2003, doi: 10.1145/937503.937505.
- [245] K. K. H. Ng, C. K. M. Lee, F. T. S. Chan, and Y. Lv, “Review on meta-heuristics approaches for airside operation research,” *Appl Soft Comput*, vol. 66, pp. 104–133, May 2018, doi: 10.1016/j.asoc.2018.02.013.
- [246] Q. Ge, Q. Yang, P. Zhuo, G. Liu, and S. Tang, “Genetic Algorithm-Based Sensor Allocation With Nonlinear Centralized Fusion Observable Degree,” *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 12, pp. 3665–3673, Dec. 2019, doi: 10.1109/TNNLS.2019.2918220.
- [247] O. A. W. M. Fogel LJ, “Artificial Intelligence through Simulated Evolution,” in *Evolutionary Computation*, IEEE, 1966. doi: 10.1109/9780470544600.ch7.
- [248] Rechenberg I, “Evolutions strategie: optimierung technischer Systeme nach Prinzipien der biologischen Evolution,” *frommann-holzboog, Stuttgart, Germany*, 1973.
- [249] J. H. Holland, “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence,” *MIT press*, 1992.
- [250] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Comput Oper Res*, vol. 13, no. 5, pp. 533–549, Jan. 1986, doi: 10.1016/0305-0548(86)90048-1.
- [251] T.-C. Ou, W.-F. Su, X.-Z. Liu, S.-J. Huang, and T.-Y. Tai, “A Modified Bird-Mating Optimization with Hill-Climbing for Connection Decisions of Transformers,” *Energies (Basel)*, vol. 9, no. 9, p. 671, Aug. 2016, doi: 10.3390/en9090671.
- [252] D. L. Poole and A. K. Mackworth, *Artificial Intelligence*. Cambridge University Press, 2023. doi: 10.1017/9781009258227.
- [253] S. S. Skiena, *The Algorithm Design Manual*. London: Springer London, 2008. doi: 10.1007/978-1-84800-070-4.
- [254] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science (1979)*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.
- [255] I. H. Osman and G. Laporte, “Metaheuristics: A bibliography,” *Ann Oper Res*, vol. 63, no. 5, pp. 511–623, Oct. 1996, doi: 10.1007/BF02125421.
- [256] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Comput Intell Mag*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.
-

-
- [257] J. Schwarzrock, I. Zacarias, A. L. C. Bazzan, R. Q. de Araujo Fernandes, L. H. Moreira, and E. P. de Freitas, “Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence,” *Eng Appl Artif Intell*, vol. 72, pp. 10–20, Jun. 2018, doi: 10.1016/j.engappai.2018.03.008.
- [258] M. Š. A. & N. G. Hüttenrauch, “Deep reinforcement learning for swarm systems,” *Journal of Machine Learning Research*, vol. 20 (54), pp. 1–31, 2019.
- [259] P. D. Andina D, *Computational intelligence: for engineering and manufacturing*. Boston, MA: Springer US, 2007. doi: 10.1007/0-387-37452-3.
- [260] M. G. Ball, B. Qela, and S. Wesolkowski, “A Review of the Use of Computational Intelligence in the Design of Military Surveillance Networks,” 2016, pp. 663–693. doi: 10.1007/978-3-319-26450-9_24.
- [261] Engelbrecht AP, *Fundamentals of computational swarm intelligence*. Wiley and Sons Ltd, Chichester, 2005.
- [262] Mataric MJ, “Interaction and intelligent behavior ,” *Massachusetts Institute of Tech, Cambridge Artificial Intelligence Lab*, 1994.
- [263] Y. Tao, “Swarm intelligence in humans: A perspective of emergent evolution,” *Physica A: Statistical Mechanics and its Applications*, vol. 502, pp. 436–446, Jul. 2018, doi: 10.1016/j.physa.2018.02.120.
- [264] S. Garnier, J. Gautrais, and G. Theraulaz, “The biological principles of swarm intelligence,” *Swarm Intelligence*, vol. 1, no. 1, pp. 3–31, Oct. 2007, doi: 10.1007/s11721-007-0004-y.
- [265] C. Blum and X. Li, “Swarm Intelligence in Optimization,” in *Swarm Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 43–85. doi: 10.1007/978-3-540-74089-6_2.
- [266] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, IEEE, 2007, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.
- [267] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, “The Bees Algorithm — A Novel Tool for Complex Optimisation Problems,” in *Intelligent Production Machines and Systems*, Elsevier, 2006, pp. 454–459. doi: 10.1016/B978-008045157-2/50081-X.
- [268] P. Bailis, R. Nagpal, and J. Werfel, “Positional Communication and Private Information in Honeybee Foraging Models,” 2010, pp. 263–274. doi: 10.1007/978-3-642-15461-4_23.
- [269] D. Karaboga and B. Akay, “A comparative study of Artificial Bee Colony algorithm,” *Appl Math Comput*, vol. 214, no. 1, pp. 108–132, Aug. 2009, doi: 10.1016/j.amc.2009.03.090.
- [270] H. F. Wedde, M. Farooq, and Y. Zhang, “BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior,” vol. 3175, Springer, 2004, pp. 83–94. doi: 10.1007/978-3-540-28646-2_8.
- [271] D. & D. M. Teodorovic, “Bee colony optimization—a cooperative learning approach to complex transportation problems.,” *Advanced OR and AI methods in transportation*, pp. 51–60, 2005.
- [272] Yang XS, “Engineering optimizations via nature-inspired virtual bee algorithms,” in *International Work-Conference on the Interplay Between Natural and Artificial Computation*, Berlin, Heidelberg: Springer, 2005, pp. 317–323.
- [273] A. Afshar, O. Bozorg Haddad, M. A. Mariño, and B. J. Adams, “Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation,” *J Franklin Inst*, vol. 344, no. 5, pp. 452–462, Aug. 2007, doi: 10.1016/j.jfranklin.2006.06.001.
- [274] C. S. D. M. M. J. & G. B. Jones, “Principled approaches to the design of multi-robot systems,” in *Proc. of the Workshop on Networked Robot*, IEEE/RSJ International Conference on Intelligent Robots and Systems , 2004.
-

-
- [275] Smith, “The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver,” *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, Dec. 1980, doi: 10.1109/TC.1980.1675516.
- [276] S. Banerjee and J. P. Hecker, “A Multi-agent System Approach to Load-Balancing and Resource Allocation for Distributed Computing,” 2017, pp. 41–54. doi: 10.1007/978-3-319-45901-1_4.
- [277] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman, “Tycoon: An implementation of a distributed, market-based resource allocation system,” *Multiagent and Grid Systems*, vol. 1, no. 3, pp. 169–182, Oct. 2005, doi: 10.3233/MGS-2005-1303.
- [278] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan, “Analyzing Market-Based Resource Allocation Strategies for the Computational Grid,” *Int J High Perform Comput Appl*, vol. 15, no. 3, pp. 258–281, Aug. 2001, doi: 10.1177/109434200101500305.
- [279] Q. Baert, A. C. Caron, M. Morge, and J.-C. Routier, “Fair Multi-agent Task Allocation for Large Data Sets Analysis,” vol. 54, 2016, pp. 24–35. doi: 10.1007/978-3-319-39324-7_3.
- [280] E. G. Jones, M. Bernardine Dias, and A. Stentz, “Learning-enhanced market-based task allocation for oversubscribed domains,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2007, pp. 2308–2313. doi: 10.1109/IROS.2007.4399534.
- [281] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-Based Multirobot Coordination: A Survey and Analysis,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, Jul. 2006, doi: 10.1109/JPROC.2006.876939.
- [282] D.-H. Lee, “Resource-based task allocation for multi-robot systems,” *Rob Auton Syst*, vol. 103, pp. 151–161, May 2018, doi: 10.1016/j.robot.2018.02.016.
- [283] M. Nanjanath and M. Gini, “Repeated auctions for robust task execution by a robot team,” *Rob Auton Syst*, vol. 58, no. 7, pp. 900–909, Jul. 2010, doi: 10.1016/j.robot.2010.03.011.
- [284] R. Zlot and A. Stentz, “Market-based Multirobot Coordination for Complex Tasks,” *Int J Rob Res*, vol. 25, no. 1, pp. 73–101, Jan. 2006, doi: 10.1177/0278364906061160.
- [285] S. C. Botelho and R. Alami, “M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, IEEE, 1999, pp. 1234–1239. doi: 10.1109/ROBOT.1999.772530.
- [286] S. C. Botelho and R. Alami, “Multi-robot cooperation through the common use of ‘mechanisms,’” in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, Maui, HI, USA: IEEE, 2001, pp. 375–380. doi: 10.1109/IROS.2001.973386.
- [287] B. P. Gerkey and M. J. Mataric, “MURDOCH: publish/subscribe task allocation for heterogeneous agents,” in *Proceedings of the fourth international conference on Autonomous agents*, New York, NY, USA: ACM, Jun. 2000, pp. 203–204. doi: 10.1145/336595.337369.
- [288] B. P. Gerkey and M. J. Mataric, “Sold!: auction methods for multirobot coordination,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, Oct. 2002, doi: 10.1109/TRA.2002.803462.
- [289] M. B. Dias, “TraderBots: a new paradigm for robust and efficient multirobot coordination in dynamic environments,” Carnegie Mellon University, 2004.
- [290] M. Bernardine Dias, M. Zinck, R. Zlot, and A. Stentz, “Robust multirobot coordination in dynamic environments,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, IEEE, 2004, pp. 3435–3442 Vol.4. doi: 10.1109/ROBOT.2004.1308785.
-

-
- [291] S. D. P. M. F. A. J. N. & T. C. Ramchurn, “Coalition formation with spatial and temporal constraints,” in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Toronto, Canada, 2010, pp. 1181–1188.
- [292] K. Macarthur, R. Stranders, S. Ramchurn, and N. Jennings, “A Distributed Anytime Algorithm for Dynamic Task Allocation in Multi-Agent Systems,” *Proceedings of the AAI Conference on Artificial Intelligence*, vol. 25, no. 1, pp. 701–706, Aug. 2011, doi: 10.1609/aaai.v25i1.7866.
- [293] S. D. Ramchurn, A. Farinelli, K. S. Macarthur, and N. R. Jennings, “Decentralized Coordination in RoboCup Rescue,” *Comput J*, vol. 53, no. 9, pp. 1447–1461, Nov. 2010, doi: 10.1093/comjnl/bxq022.
- [294] H. Yedidsion, R. Zivan, and A. Farinelli, “Applying max-sum to teams of mobile sensing agents,” *Eng Appl Artif Intell*, vol. 71, pp. 87–99, May 2018, doi: 10.1016/j.engappai.2018.02.017.
- [295] W. E. Walsh and M. P. Wellman, “A market protocol for decentralized task allocation,” in *Proceedings International Conference on Multi Agent Systems (Cat. No.98EX160)*, IEEE Comput. Soc, 1998, pp. 325–332. doi: 10.1109/ICMAS.1998.699077.
- [296] N. Kalra, D. Ferguson, and A. Stentz, “A Generalized Framework for Solving Tightly-coupled Multirobot Planning Problems,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy: IEEE, Apr. 2007, pp. 3359–3364. doi: 10.1109/ROBOT.2007.363991.
- [297] A. R. Mosteo, L. Montano, and M. G. Lagoudakis, “Multi-robot routing under limited communication range,” in *2008 IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, USA: IEEE, May 2008, pp. 1531–1536. doi: 10.1109/ROBOT.2008.4543419.
- [298] J. & O. G. Guerrero, “Multi-robot task allocation method for heterogeneous tasks with priorities,” in *In Distributed Autonomous Robotic Systems 6*, Tokyo, Japan: Springer, 2007, pp. 181–190.
- [299] M. Nanjanath and M. Gini, “Dynamic task allocation for robots via auctions,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, USA: IEEE, 2006, pp. 2781–2786. doi: 10.1109/ROBOT.2006.1642122.
- [300] Fang Tang and L. E. Parker, “ASyMTRe: Automated Synthesis of Multi-Robot Task Solutions through Software Reconfiguration,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain: IEEE, 2005, pp. 1501–1508. doi: 10.1109/ROBOT.2005.1570327.
- [301] S. Amador, S. Okamoto, and R. Zivan, “Dynamic Multi-Agent Task Allocation with Spatial and Temporal Constraints,” *Proceedings of the AAI Conference on Artificial Intelligence*, vol. 28, no. 1, pp. 1495–1496, Jun. 2014, doi: 10.1609/aaai.v28i1.8889.
- [302] P. Lucic and D. Teodorovic, “Transportation modeling: an artificial life approach,” in *14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings.*, IEEE Comput. Soc, 2002, pp. 216–223. doi: 10.1109/TAI.2002.1180807.
- [303] M. Dorigo, V. Maniezzo, and A. Coloni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, Feb. 1996, doi: 10.1109/3477.484436.
- [304] I. Tkach, Y. Edan, A. Jevtic, and S. Y. Nof, “Automatic Multi-sensor Task Allocation Using Modified Distributed Bees Algorithm,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Manchester, UK: IEEE, Oct. 2013, pp. 1401–1406. doi: 10.1109/SMC.2013.242.
- [305] M. Otte, M. J. Kuhlman, and D. Sofge, “Auctions for multi-robot task allocation in communication limited environments,” *Auton Robots*, vol. 44, no. 3–4, pp. 547–584, Mar. 2020, doi: 10.1007/s10514-019-09828-5.
-

-
- [306] D. Tihanyi, Y. Lu, O. Karaca, and M. Kamgarpour, “Multi-robot task allocation for safe planning against stochastic hazard dynamics,” in *2023 European Control Conference (ECC)*, IEEE, Jun. 2023, pp. 1–6. doi: 10.23919/ECC57647.2023.10178126.
- [307] C. Pinciroli *et al.*, “ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems,” *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012, doi: 10.1007/s11721-012-0072-5.
- [308] M. Hutter *et al.*, “ANYmal - toward legged robots for harsh environments,” *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, Sep. 2017, doi: 10.1080/01691864.2017.1378591.
- [309] S. Michieletto, S. Ghidoni, E. Pagello, M. Moro, and E. Menegatti, “Why teach robotics using ROS,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, pp. 60–68, Jan. 2014, doi: 10.14313/JAMRIS_1-2014/8.
- [310] X. Wang, X. Ma, X. Li, X. Ma, and C. Li, “Target-biased informed trees: sampling-based method for optimal motion planning in complex environments,” *J Comput Des Eng*, vol. 9, no. 2, pp. 755–771, Apr. 2022, doi: 10.1093/jcde/qwac025.
- [311] S. Lee, E. Milner, and S. Hauert, “A Data-Driven Method for Metric Extraction to Detect Faults in Robot Swarms,” *IEEE Robot Autom Lett*, vol. 7, no. 4, pp. 10746–10753, Oct. 2022, doi: 10.1109/LRA.2022.3189789.
- [312] S. H. Kim, S.-G. Yoon, S. H. Chae, and S. Park, “Economic and environmental optimization of a multi-site utility network for an industrial complex,” *J Environ Manage*, vol. 91, no. 3, pp. 690–705, Jan. 2010, doi: 10.1016/j.jenvman.2009.09.033.
- [313] K.-H. Lee, O. Nachum, T. Zhang, S. Guadarrama, J. Tan, and W. Yu, “PI-ARS: Accelerating Evolution-Learned Visual-Locomotion with Predictive Information Representations,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2022, pp. 1447–1454. doi: 10.1109/IROS47612.2022.9981952.
- [314] A. Das, O. Naroditsky, Zhiwei Zhu, S. Samarasekera, and R. Kumar, “Robust visual path following for heterogeneous mobile platforms,” in *2010 IEEE International Conference on Robotics and Automation*, IEEE, May 2010, pp. 2431–2437. doi: 10.1109/ROBOT.2010.5509699.
- [315] M. Debord, W. Honig, and N. Ayanian, “Trajectory Planning for Heterogeneous Robot Teams,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2018, pp. 7924–7931. doi: 10.1109/IROS.2018.8593876.
- [316] Z. G. Saribatur, V. Patoglu, and E. Erdem, “Finding optimal feasible global plans for multiple teams of heterogeneous robots using hybrid reasoning: an application to cognitive factories,” *Auton Robots*, vol. 43, no. 1, pp. 213–238, Jan. 2019, doi: 10.1007/s10514-018-9721-x.
- [317] J. Collins, S. Chand, A. Vanderkop, and D. Howard, “A Review of Physics Simulators for Robotic Applications,” *IEEE Access*, vol. 9, pp. 51416–51431, 2021, doi: 10.1109/ACCESS.2021.3068769.
- [318] Y. Yu, Z. Miao, X. Wang, and L. Shen, “Cooperative circumnavigation control of multiple unicycle-type robots with non-identical input constraints,” *IET Control Theory & Applications*, vol. 16, no. 9, pp. 889–901, Jun. 2022, doi: 10.1049/cth2.12275.
- [319] C. Calderón-Arce, J. C. Brenes-Torres, and R. Solis-Ortega, “Swarm Robotics: Simulators, Platforms and Applications Review,” *Computation*, vol. 10, no. 6, p. 80, May 2022, doi: 10.3390/computation10060080.
- [320] M. U. Arif and S. Haider, “A Flexible Framework for Diverse Multi-Robot Task Allocation Scenarios Including Multi-Tasking,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 16, no. 1, pp. 1–23, Mar. 2021, doi: 10.1145/3502200.
-

-
- [321] P. Fraisse, D. Andreu, R. Zapata, J. P. Richard, and T. Divoux, "Remote decentralized control strategy for cooperative mobile robots," in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, IEEE, 2004, pp. 1011–1016. doi: 10.1109/ICARCV.2004.1468982.
- [322] M. Dihya, M. Moufid, B. Chemseddine, and B. Moussaab, "Switched time delay control based on neural network for fault detection and compensation in robot," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 10, no. 2, p. 91, Jun. 2021, doi: 10.11591/ijra.v10i2.pp91-103.
- [323] W. Łabuński and A. Burghardt, "Software for the Control and Monitoring of Work of a Collaborative Robot," *Journal of Automation, Mobile Robotics and Intelligent Systems*, pp. 29–36, May 2022, doi: 10.14313/JAMRIS/3-2021/16.
- [324] Y.-C. Wu, J.-W. Lee, and H.-C. Wang, "Robots for search site monitoring, suspect guarding, and evidence identification," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 9, no. 2, p. 84, Jun. 2020, doi: 10.11591/ijra.v9i2.pp84-93.
- [325] Ayman. El shenawy, Khalil. Mohamed, and Hany. M. Harb, "Exploration Strategies of Coordinated Multi-Robot System: A Comparative Study," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 7, no. 1, p. 48, Mar. 2018, doi: 10.11591/ijra.v7i1.pp48-58.
- [326] M. Stolzle, T. Miki, L. Gerdes, M. Azkarate, and M. Hutter, "Reconstructing Occluded Elevation Information in Terrain Maps With Self-Supervised Learning," *IEEE Robot Autom Lett*, vol. 7, no. 2, pp. 1697–1704, Apr. 2022, doi: 10.1109/LRA.2022.3141662.
- [327] H. S. Dakheel, Z. B. Abdullah, and S. W. Shneen, "Advanced optimal GA-PID controller for BLDC motor," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 2077–2086, Aug. 2023, doi: 10.11591/eei.v12i4.4649.
- [328] P. D. Kusuma and M. Kallista, "Pickup and delivery problem in the collaborative city courier service by using genetic algorithm and nearest distance," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 2, pp. 1026–1036, Apr. 2022, doi: 10.11591/eei.v11i2.3223.
- [329] I. Ahmed Abed, M. Mohammed Ali, and A. Abood Abdul Kadhim, "Using particle swarm optimization to solve test functions problems," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 6, pp. 3422–3431, Dec. 2021, doi: 10.11591/eei.v10i6.3244.
- [330] A. Elbatal, A. M. Youssef, and M. M. Elkhatib, "Smart aerosonde UAV longitudinal flight control system based on genetic algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2433–2441, Oct. 2021, doi: 10.11591/eei.v10i5.2342.
- [331] L. Joseph and J. Cacace, *Mastering ROS for robotics programming: design, build, and simulate complex robots using Robot Operating System*, no. 1. 2015. [Online]. Available: www.packtpub.com
- [332] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*, vol. 1. Addison-wesley Reading, 1999.
- [333] X. Wang, X. Ma, X. Li, X. Ma, and C. Li, "Target-biased informed trees: sampling-based method for optimal motion planning in complex environments," *J Comput Des Eng*, vol. 9, no. 2, pp. 755–771, 2022, doi: 10.1093/jcde/qwac025.

Annexes

Annexes

Annexe A : La conception électrique du vLiDAR3D

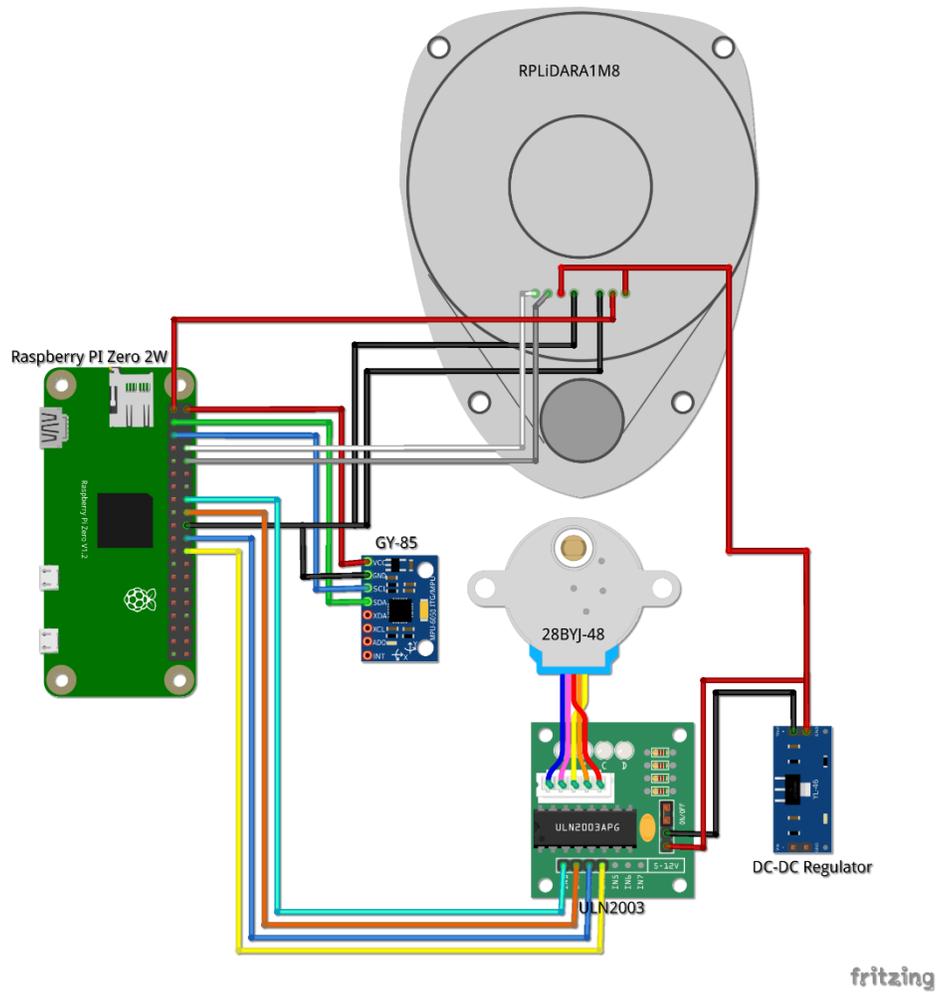
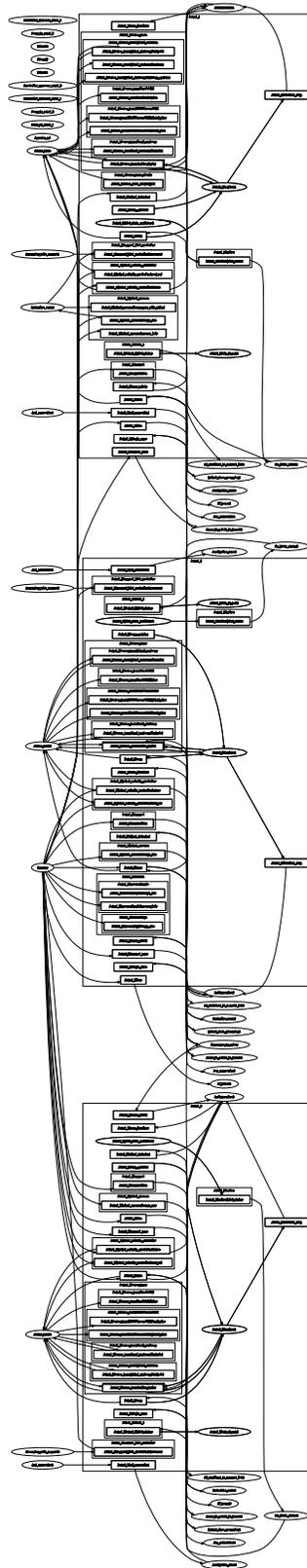


Schéma de brochage du vLiDAR3D (développé sous Fritzing)

Annexe B : Le ros_graph (le graphe de nœuds de ROS qui ont en exécution) de la simulation de collaboration de trois robots



ملخص

إن معدل زيادة الأخطار الطبيعية والكوارث التي تتسبب فيها المجتمعات البشرية، يدفع نحو الحاجة إلى حلول عاجلة وفعالة. فالحوادث التي تنشأ خلال هذه المخاطر تستلزم اللجوء إلى عمليات البحث والإنقاذ كمنافرة أولى. تتطلب التدخلات في المواقع المتضررة إدخال استراتيجيات تعتمد على مسح القطاعات. يوفر نشر أنظمة الروبوتات الدقيقة وسيلة واعدة لتحسين قدرات الاستجابة في هذه البيئات المعادية. وبطبيعة الحال، يتعاون كل عامل من الروبوتات الدقيقة ويتعاون مع المجموعة الكاملة التي تم تجميعها لغرض هذا النوع من المهام. تستكشف هذه الأطروحة مدى صلاحية المساهمة في تطوير مجموعة من الروبوتات الدقيقة التي تم تكييفها خصيصاً للنشر في المواقع المعادية. بالاستفادة من التقدم في مجال الروبوتات وتقنيات الاتصالات وتكامل إنترنت الأشياء بالإضافة إلى حزم نشر شبكة إنترنت الأشياء الفعالة والأمنة. تستكشف دراستنا السلوكيات التعاونية واستراتيجيات الملاحة والفعالية التشغيلية لعوامل الروبوتات الدقيقة هاته في البيئات الصعبة ذات الخصائص الديناميكية. وبشكل التحليل الشامل للمقاربات الخوارزمية وتقنيات الإرسال وأمن الشبكة ومنهجيات المحاكاة والتحقق التجريبي العمود الفقري لهذا البحث، مما يوفر رؤية قيمة حول إمكانات الروبوتات الدقيقة لحل جهود الاستجابة للكوارث. وتسلط النتائج الضوء على أهمية مجموعات الروبوتات الصغيرة الموجهة نحو إنترنت الأشياء لزيادة الوعي الظرفي والسلامة، وتسهيل تبادل البيانات في الوقت الحقيقي وتمكين التعاون السلس بين الكيانات الروبوتية. وفي الوقت الذي يستعد فيه العالم لمستقبل لا يمكن التنبؤ به بشكل متزايد، تساهم التطورات المعروضة في هذه الأطروحة في التطور المستمر للأنظمة الروبوتية لمهام الإغاثة الإنسانية والإغاثة في حالات الكوارث، وتهدف في نهاية المطاف إلى إنشاء مجتمعات أكثر أماناً ومرونة. الكلمات المفتاحية: الروبوت الصغير، الكتلة، المواقع المعادية، الرقمنة ثلاثية الأبعاد، نظام تشغيل الروبوت (ROS)، الخوارزمية التعاونية.

Résumé

Le taux d'accroissement des aléas naturels et des catastrophes causées par les sociétés humaines, induit un besoin de solutions urgentes et effectives. Les incidents qui surgissent durant ces aléas implique un recours vers les opérations de recherche et de sauvetage en tant que première manœuvre. Les interventions sur les sites affectés nécessitent l'introduction de stratégies basées sur le balayage par segments. A cet effet, le déploiement de systèmes de microrobots présente une voie prometteuse pour améliorer les capacités de réponse sur ces environnements hostiles. Certes, chaque agent microrobot collabore et coopère avec tout le cluster monté pour le besoin de ce type de mission. Cette thèse explore la validité de la contribution au développement d'un cluster de micro-cobots spécifiquement adaptés au déploiement sur des sites hostiles. Tirant parti des progrès de la robotique, des technologies de communication et de l'intégration de l'IoT ainsi que de conditionnement de déploiement de réseaux IoT effectifs et sécurisés. Notre étude explore les comportements collaboratifs, les stratégies de navigation et l'efficacité opérationnelle de ces agents microrobots dans des environnements difficiles à caractères dynamiques. Une analyse complète des approches algorithmiques, des technologies de transmission, de la sécurité des réseaux, des méthodologies de simulation et des validations expérimentales constituent l'épine dorsale de cette recherche, fournissant des informations précieuses sur le potentiel des microcobots à solutionner les efforts de réponse aux catastrophes. Les résultats mettent en évidence l'importance des clusters de microcobots orientés IoT pour accroître la connaissance de la situation et la sécurité, faciliter l'échange de données en temps réel et permettre une collaboration transparente entre les entités robotiques. Alors que le monde se prépare à un avenir de plus en plus imprévisible, les développements présentés dans cette thèse contribuent à l'évolution continue des systèmes robotiques pour les missions humanitaires et de secours en cas de catastrophe, visant à terme à créer des communautés plus sûres et plus résilientes.

Mots-clés : Micro-cobot, Cluster, Sites hostiles, Digitalisation 3D, Système d'exploitation pour robots (ROS), Algorithme collaboratif.

Abstract

The escalating frequency of natural hazards and disasters resulting from human activity necessitates prompt and efficient solutions. Search and rescue operations are typically the initial response to such incidents. Strategies based on segment scanning are required for interventions at affected sites. In this regard, the use of microrobot systems shows potential for enhancing response capabilities in these hostile environments. Each microrobot agent collaborates and cooperates with the entire cluster assembled for the mission. This thesis examines the development of a cluster of micro-cobots adapted for deployment on hostile sites. The study takes advantage of advances in robotics, communication technologies, and IoT integration, as well as effective and secure IoT network deployment packaging. The study examines the collaborative behaviors, navigation strategies, and operational efficiency of microrobot agents in challenging and dynamic environments. The research is based on a comprehensive analysis of algorithmic approaches, transmission technologies, network security, simulation methodologies, and experimental validations. The findings provide valuable insights into the potential of microrobots to aid in disaster response efforts. The findings emphasize the significance of microcobot clusters oriented towards IoT in enhancing situational awareness and safety, enabling real-time data exchange, and promoting seamless collaboration among robotic entities. With the ultimate goal of creating safer and more resilient communities, this thesis presents developments that contribute to the ongoing evolution of robotic systems for humanitarian and disaster relief missions.

Keywords: Micro-cobot, Cluster, Hostile Sites, 3D Digitization, Robotics Operating System (ROS), Collaborative algorithm.