



HAL
open science

Reconnaissance automatisée de points d'intérêts pour un robot d'inspection dans un environnement contraint et dégradé : inspection visuelle et chimique par un robot hexapode

Guillaume Morin-Duponchelle

► To cite this version:

Guillaume Morin-Duponchelle. Reconnaissance automatisée de points d'intérêts pour un robot d'inspection dans un environnement contraint et dégradé : inspection visuelle et chimique par un robot hexapode. Automatique / Robotique. ENSTA Bretagne - École nationale supérieure de techniques avancées Bretagne, 2022. Français. NNT : 2022ENTA0009 . tel-04297138

HAL Id: tel-04297138

<https://theses.hal.science/tel-04297138>

Submitted on 21 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *ROBEX : Robotics for exploration*

Par

« Guillaume MORIN-DUPONCHELLE »

« Reconnaissance automatique de points d'intérêts dans un environnement contraint et dégradé »

« Inspection visuelle et chimique par un robot hexapode. »

Thèse présentée et soutenue à ENSTA Bretagne, Brest, le 12 Décembre 2022

Unité de recherche : « Equipe PRASYS "Perception, Robotique, Autonomous SYStems" »

Rapporteurs avant soutenance :

M. Vincent HUGEL Professeur des universités, Université de Toulon
M. Abderraouf BENALI Professeur des universités, Université de Versailles Saint-Quentin-en-Yvelines

Composition du Jury :

Président :	M. Luc JAULIN	Professeur des universités, ENSTA Bretagne
Examineurs :	M. David BAUDRY	Enseignant Chercheur, HDR, directeur de recherche à CESI LINEACT
	M. Lamri NEHAOUA	Maître de conférences, Université Évy Paris-Saclay
Dir. de thèse :	M. Benoit ZERR	Enseignant Chercheur, Maître de conférence HDR, ENSTA Bretagne Brest
Encadrant de thèse :	M. Ahmed NAIT CHABANE	Enseignant chercheur, CESI Lineact

Invitée :

Mme. Aurélie GUESDON Ingénieur scientifique. Pilgrim Technology

REMERCIEMENTS

Il me sera très difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme. Je voudrais tout d'abord remercier grandement mon directeur de thèse, Benoit ZERR, mon encadrant de thèse, Ahmed NAIT CHABANE, ainsi que la directrice de Pilgrim Technology, Anne-Marie HAUTE, mon encadrante technique Aurélie GUESDON, pour leurs aides tout au long de ce projet de thèse. Je suis ravi d'avoir travaillé en leurs compagnies. Outre leurs appuis scientifiques, techniques et expertise métier, ils ont toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse. Ce travail n'aurait pas été possible sans le soutien et le partenariat de Pilgrim Technology, l'ENSTA Bretagne et le CESI LINEACT, à l'origine du projet de thèse, et qui a fortement inspirer le sujet et le titre de cette thèse. Je tiens à remercier l'aide et l'appui de mon collègue Pierre SCHOESETTERS, Ingénieur Bureau d'Etude Pilgrim Technology, pour son expertise logiciel et son professionnalisme lors des phases clés du projet. Ainsi que mes collègues qui ont contribué à l'élaboration de ArIA Inspector, Elie DE FONCLARE, Ingénieur Bureau d'Etudes, Anicet SOULARD, Dessinateur-Projeteur. Je tiens à remercier chaleureusement, mes stagiaires et apprentis en développement, qui ont contribué à la réussite de ce projet et que j'ai eu plaisir à encadrer, Geoffroy FOUGERE, Mélanie DILIGENT, Esther BAILLOUX, Remy MOUNIER. M. Vincent HUGEL et M. Abderraouf BENALI m'ont fait l'honneur d'être rapporteurs de ma thèse, ils ont pris le temps de m'écouter et de discuter avec moi. Leurs remarques m'ont permis d'envisager mon travail sous un autre angle. Pour tout cela je les remercie. Je tiens à remercier M. Luc JAULIN et M. David BAUDRY pour avoir accepté de participer à mon jury de thèse et pour leurs participations scientifiques ainsi que le temps qu'il ont consacré à ma recherche. Je remercie également M. Lamri NEHAOUA pour l'honneur qu'il me fait d'être dans mon jury de thèse. Je remercie toutes les personnes avec qui j'ai partagé mes études et notamment ces années de thèse. Je remercie aussi Christian Vanderriest, pour sa passion de la science.

TABLE OF CONTENTS

1	Introduction	23
1.1	L'inspection robotisée	24
1.1.1	Contexte de la thèse	25
1.1.2	Problématiques et objectifs de la thèse	27
1.1.3	Organisation de la thèse	32
1.2	Étude des éléments contextuels	33
1.2.1	Inspection des milieux contraints	34
1.2.2	Les milieux contraints et dégradés	34
1.2.3	Les critères d'évaluation	38
1.2.4	Évaluation des cas d'études	44
1.2.5	Synthèse	49
1.2.6	Expression des besoins	51
1.3	Conclusion de fin de chapitre	53
2	État de l'art	54
2.1	État de l'art : Inspection industrielle	55
2.1.1	Introduction	55
2.1.2	Les méthodes d'inspections visuelles	55
2.1.3	Synthèse générale : choix des capteurs	59
2.2	État de l'art : La plateforme robotique terrestre	61
2.2.1	Introduction	61
2.2.2	La notation	62
2.2.3	Synthèse générale : Catégorie de robot	71
2.3	État de l'art : les robots hexapodes	72
2.3.1	Introduction	73
2.3.2	Architecture des hexapodes	75
2.3.3	Synthèse générale : Le choix d'architecture	78
2.4	État de l'art : la détection des points d'intérêts visuels	79
2.4.1	Les catégories de méthodes	80

TABLE OF CONTENTS

2.4.2	Évaluation de l'architecture de détection des points d'intérêts visuels	86
2.4.3	Synthèse générale : choix de la méthode visuelle	87
2.5	État de l'art : la détection de points intérêts chimiques	88
2.5.1	Introduction	88
2.5.2	Les méthodes de simulation	88
2.5.3	Asservissement chimique : La localisation des sources de gaz	89
2.5.4	L'asservissement chimique d'un robot hexapode	90
2.5.5	Synthèse générale : choix de la méthode chimique	93
2.6	Conclusion de fin de chapitre	93
3	Méthodologie	95
3.1	Approche proposée	95
3.1.1	Les problématiques de l'approche	95
3.1.2	Approche globale	98
3.2	Brique de contrôle de l'hexapode	100
3.2.1	Système de contrôle d'un robot hexapode	101
3.3	Brique de navigation d'un robot hexapode	112
3.3.1	L'odométrie	113
3.3.2	AMCL : Localisation par adaptation Monte Carlo	115
3.3.3	Gestion de la planification de trajectoire	116
3.4	Brique d'asservissement visuel	118
3.4.1	Asservissement visuel d'un robot hexapode	118
3.4.2	La détection automatisée de points d'intérêts	131
3.5	Brique d'asservissement chimique	132
3.5.1	Asservissement chimique d'un robot hexapode	133
3.6	Conclusion de fin de chapitre	136
4	Implémentation et intégration	139
4.1	Application de l'approche	140
4.1.1	Architecture du robot	141
4.2	Architecture du contrôleur	149
4.2.1	Méthode de Réglage PID : Simulation	152
4.3	Architecture de navigation	152
4.3.1	Implémentation de la navigation	153
4.4	Asservissement visuel	156

4.4.1	Application de l’asservissement visuel par le QR Code	156
4.4.2	Application de l’asservissement visuel par la reconnaissance d’objets	157
4.5	Asservissement chimique	160
4.5.1	L’approche chimique	160
4.5.2	L’environnement de simulation	161
4.6	Conclusion de fin de chapitre	163
5	Expérimentations et Résultats	165
5.0.1	Introduction	165
5.1	Expérimentations en simulation	166
5.1.1	Protocole expérimentale pour la simulation	166
5.1.2	Expérimentation et résultats de l’odométrie	167
5.1.3	Expérimentation et résultat de la navigation	170
5.1.4	Expérimentations et résultats de l’asservissement visuel	171
5.1.5	Entraînement et résultats de YOLO V5	177
5.1.6	Expérimentation et résultat de l’asservissement chimique	185
5.2	Expérimentations sur le robot réel	192
5.2.1	Expérimentation et résultats des essais statiques	193
5.2.2	Résultats en état dynamique	196
5.2.3	Expérimentation et résultats du franchissement d’obstacles	200
5.2.4	Conclusion	202
5.3	Présentation de scénarios représentatifs	203
5.3.1	Simulation chimique : Labyrinthe	203
5.3.2	Essai réalisé à l’entreprise Naval Group	204
5.4	Conclusion de fin de chapitre	208
6	Conclusions : contributions et perspectives	211
6.1	Contributions	211
6.2	Discussions	212
6.3	Perspectives de l’étude	215
7	Annexes	217
7.1	ANNEXE A : Capteurs de gaz	217
7.2	ANNEXE B : Logigramme de l’approche	218

TABLE OF CONTENTS

7.3	ANNEXE C : Fichier de configuration (json) de la cinématique d'ArIA Inspector	219
7.4	ANNEXE D : Réglage PID : Génération des paramètres (Pid.cfg)	222
7.5	ANNEXE E : Fonction réglage automatique des PID.	223
7.6	ANNEXE F : Architecture de YOLO V5	224
8	Bibliographie	226
9	Résumé étendu	251

TABLE DES FIGURES

1.1	Air-Cobot (Aircraft enhanced Inspection by smaRt Collaborative rOBOT), a été développé pour contrôler la qualité des différentes pièces aéronautiques par des contrôles non-destructifs [1].	24
1.2	Organisation de l'équipe encadrant la thèse.	26
1.3	Sphère de stockage de gaz butane à Abidjan.	30
1.4	Citerne de stockage à inspecter.	31
1.5	A : Carrière souterraine de calcaire (60 hectares) à Conflans Sainte Honorine, Herblay [5]. B : Carrière et effondrement d'une maison à Casablanca suite à de forte précipitation [30].	31
1.6	Mine de fer en Lorraine [104].	36
1.7	Image aérienne d'un séisme de magnitude 6,8 en Turquie, dans une zone urbaine [110].	37
1.8	Diagramme "Bête à cornes" pour la définition du besoin.	51
1.9	Diagramme des interactions avec les critères de l'environnement.	51
2.1	Tableau de la robotique mobile terrestre, inspiré des travaux de [115]	62
2.2	Les robots à roues pour les milieux risqués et complexes. A : Axel Rover [116], B : Venus Rover [117] [36], C : Opportunity Rover [35], D : Rover holonome avec des roues "tentacules" [118]	64
2.3	A : Nanokhod [37], B : Robot à 4 chenilles [120], C : Robot multisegments à 8 chenille : OmniTread	65
2.4	A : Micro robot à saut [122], B : BounceBot (Salto) [124]	67
2.5	A : Ascento, robot hybride roulant et à saut [126], B : ANYmal, robot hybride marchant [125]	68
2.6	A : Walk man (2 pattes) [40], B : AQuRo (4 pattes) [41], C : Weaver (6 pattes) [43], D : Spider-Robot (8 pattes) [44].	69

TABLE DES FIGURES

2.7	Différents types d'architectures d'hexapode : (A) architecture de base RHex [63]; (B) Le robot marcheur à six pattes LAURON V [66]; (C) ATHLÈTE pendant les tests expérimentaux [68]; (D) Bruce, un hexapode réalisé par l'université de Zurich [70].	76
2.8	Évaluation des récentes architectures de YOLO : YOLO4, YOLO5 et PP-YOLO [91]	85
2.9	Comparaison des outils de simulation permettant la recherche de source de gaz pour une application robotique [176].	89
2.10	Algorithme "Cast" [181].	91
2.11	Algorithme "surge-spiral" [207]	92
2.12	Algorithme "surge-cast" [205]	92
3.1	Approche proposée pour les données capteurs (en vert), le contrôle (en bleu), la navigation (en orange) et l'asservissement visuel (en rose) et chimique du robot hexapode (en violet).	99
3.2	Approche proposée pour le contrôle du robot hexapode. En bleu : la brique de contrôle, en jaune : la brique de navigation et en noir : les servomoteurs.	100
3.3	Concept de modélisation d'une patte.	102
3.4	Modélisation d'une patte d'hexapode avec les systèmes de coordonnées. . .	103
3.5	Démarche d'un robot hexapode des travaux de Tam et al. [211].	110
3.6	Approche proposée pour la navigation du robot hexapode. En jaune : la brique de navigation, en bleu : le contrôleur de l'hexapode et en vert : les données capteur	113
3.7	Méthode d'asservissement visuel. En rose : la brique d'asservissement visuel, en jaune : la brique de navigation et en vert : les données des capteurs.	119
3.8	Principe de fonctionnement du QR code.	122
3.9	Jacobien d'image [140]	122
3.10	L'image A représente un détection automatique avec une boîte englobante de l'objet détecté en vue initial et l'image B représente la vue désirée du même objet (défaut industriel).	125
3.11	Déplacement d'un robot holonome dans un repère cartésien.	126
3.12	Les images d'entrées sont découpées en régions où un objet est susceptible d'apparaître (extraction des caractéristiques). Ces régions sont ensuite classées par le réseau de neurones convolutif [203].	132

3.13	Méthode d’asservissement chimique. En violet : la brique d’asservissement chimique, en jaune : la brique de navigation et en vert : les données des capteurs.	133
3.14	Un dégagement de gaz est modélisé comme une séquence de bouffées, chacune est composée de plusieurs filaments (étant un filament une distribution normale 3D des molécules de gaz). Les bouffées sont affectées par l’advection et diffusion, altérant l’emplacement des filaments par l’effet du vent (V_a) et des processus aléatoires (V_m), ainsi que leur taille (V_d) [176].	135
4.1	Logigramme de l’approche de résolution pour un procédé d’asservissement visuel et chimique	141
4.2	Architecture du robot théorique : le robot hexapode.	142
4.3	Composition d’une patte.	142
4.4	ArIA AX18 : 1ère version	143
4.5	ArIA LUCAN, schéma de conception de la deuxième version.	144
4.6	ArIA LUCAN.	145
4.7	ArIA Inspector.	146
4.8	Spécification de ArIA Inspector.	147
4.9	Architecture de ArIA INSPECTOR.	148
4.10	Approche détaillé du contrôleur de l’hexapode OpenSHC [211]	149
4.11	Disposition des pattes d’ArIA Inspector	150
4.12	Espace de travail (bleu foncé) et Espace de marche (bleu)de ArIA Inspector.	151
4.13	Logigramme de navigation.	153
4.14	Méthode de navigation en utilisant le paradigme de ROS.	154
4.15	Asservissement visuel en utilisant un QRcode.	157
4.16	Logigramme de l’asservissement visuel par reconnaissance d’objets	158
4.17	Architecture des différentes couches de YOLO V5 (S) [242]	159
4.18	Asservissement chimique en utilisant un algorithme de recherche de gaz.	161
4.19	Architecture de traitement de l’environnement de simulation de gaz [176]	162
5.1	Illustration des différents essais opérés sous ROS et Gazebo sur l’odométrie, la navigation et la localisation, l’asservissement visuel et l’asservissement chimique.	167
5.2	Le test d’odométrie avec le robot ArIA Inspector.	168
5.3	Comparaison d’odométrie à partir d’une trajectoire.	169

TABLE DES FIGURES

5.4 Données IMU comparées aux données de vitesse souhaitées. 169

5.5 Le test de navigation avec le robot ArIA Inspector. 170

5.6 De gauche à droite : 1) le tableau de comparaison entre AMCL et la vérité terrain du robot sur la position sur l'axe X, 2) sur l'axe Y, 3) L'erreur absolue de la position sur l'axe X, 4) sur l'axe Z 171

5.7 L'asservissement visuel réalisé avec un QRcode. 172

5.8 Asservissement visuel : Trajectoire du robot sur les axes X et Y en mètres. 174

5.9 Asservissement visuel : Trajectoire du robot par rapport à la cible sur l'axe X et Y en cm. 174

5.10 Asservissement visuel : Erreur relative du robot sur les axes X et Y en cm. 175

5.11 Asservissement visuel : Répartition de la durée d'asservissement. 175

5.12 Asservissement visuel : Répartition de l'erreur absolue. 176

5.13 La matrice de corrélation suite à l'entraînement. A) La base de données utilisées, B) Empreinte des positions de l'objet, C) Corrélogramme de position et D) Corrélogramme de taille. 180

5.14 Matrice de confusion de la base de données de défauts industriels. 181

5.15 Résultats de la courbe F1 suite à l'entraînement. 182

5.16 Résultat du corrélogramme des différentes catégories de l'entraînement avec l'occurrence des classes en fonction de la largeur et la hauteur. 183

5.17 Résultat de la précision des différentes catégories de l'entraînement correctement prédites. 184

5.18 Courbe de rappel ("recall") des différentes classes de l'entraînement en fonction de l'indice de confiance. 184

5.19 Résultats de la courbe Précision Rappel sur la base de données de défauts industriels. 185

5.20 Résultats de la métrique sur l'entraînement de YOLO v5 en considérant toutes les classes. 185

5.21 Résultats visuels de l'étape de validation concernant l'entraînement de YOLO v5 sur la base de données de défauts industriels, ici les défauts sont variés et cible le métal. 186

5.22 Résultats visuels de l'étape de validation concernant l'entraînement de YOLO v5 sur la base de données de défauts industriels, ici les défauts de soudure. 187

5.23	Visualisation RViz de l'environnement "obstacle central" avec la simulation da propagation du gaz.	187
5.24	Visualisation RViz de l'environnement "maze" avec la simulation da propagation du gaz. Sur l'image de gauche, le début de l'expérimentation avec le robot au point de départ et la source de gaz matérialisé par une boite verte. Sur l'image de droite, le robot a atteint la source de propagation du gaz (boîte verte).	188
5.25	Le test d'asservissement chimique sous RViz en utilisant l'environnement Gaden [176].	189
5.26	Essai des différentes méthodes de recherche de source de gaz avec des capteurs de type métal oxydes simulés.	190
5.27	Asservissement chimique : Répartition de la durée d'asservissement.	190
5.28	Asservissement chimique : Répartition de l'erreur absolue.	191
5.29	Asservissement chimique : obstacle central.	191
5.30	Les 3 configurations différentes de garde au sol : Position A (20 cm), Position B (15 cm), Position C (25 cm).	193
5.31	Pourcentage de sollicitation d'effort sur les servomoteurs des différents segments en position statique (A)	195
5.32	Pourcentage de sollicitation d'effort sur les servomoteurs des différents segments en position statique (B)	195
5.33	Pourcentage de sollicitation d'effort sur les servomoteurs des différents segments en position statique (C)	196
5.34	Les servomoteurs Dynamixel MX28AT (Robotis) utilisés sur ArIA Inspector.	197
5.35	Sollicitation d'effort sur les servomoteurs des différents segments en marche avant en daN.m.	198
5.36	Pourcentage de sollicitation d'effort sur les servomoteurs des différents segments en marche arrière en daN.m.	198
5.37	Sollicitation d'effort sur les servomoteurs des différents segments en rotation gauche en daN.m.	199
5.38	Sollicitation d'effort sur les servomoteurs des différents segments en rotation droite en daN.m.	199
5.39	Sollicitation d'effort sur les servomoteurs des différents segments en translation gauche en daN.m.	200

TABLE DES FIGURES

5.40	Pourcentage de sollicitation d'effort sur les servomoteurs des différents segments en translation droite.	200
5.41	Franchissement d'un surbaut de 16 cm au sol en simulation sous Gazebo.	201
5.42	Franchissement d'un surbaut de 16 cm au sol en situation réelle.	201
5.43	Asservissement chimique : environnement du labyrinthe.	204
5.44	Furia Track, le robot à chenille développé par Pilgrim Technology	205
5.45	ArIA INSPECTOR, le robot hexapode développé par Pilgrim Technology	205
6.1	Nouveau design de patte prenant en compte l'effort mécanique et l'intégration des servos MX106 et MX64	216
7.1	Schéma électronique d'un capteur de dioxygène.	217
7.2	Logigramme de l'approche de résolution pour un procédé d'asservissement visuel et chimique	218

LISTE DES TABLEAUX

1.1	Tableau des critères contextuels d’environnement	39
1.2	Tableau des critères contextuels d’inspection.	40
1.3	Tableau d’évaluation des critères contextuels du risque.	41
1.4	Tableau d’évaluation des critères contextuels des percepts visuels.	42
1.5	Tableau d’évaluation des critères contextuels des percepts chimiques.	43
1.6	Tableau d’évaluation des cas sur les critères d’environnement	44
1.7	Tableau d’évaluation des cas sur les critères d’inspection.	45
1.8	Tableau d’évaluation des cas sur les critères de risque.	47
1.9	Tableau d’évaluation des cas sur les critères de percepts visuels.	47
1.10	Tableau d’évaluation des cas sur les critères de percepts chimiques.	49
2.1	Tableau de notation des différents types de plateformes de la robotique mobile	63
2.2	Evaluation des différentes architectures de robots mobiles suivant les cri- tères définis précédemment.	72
2.3	Matrice de notation des différentes approches de détection visuelle.	86
2.4	Tableau d’évaluation des architectures de détection visuel.	87
5.1	Les différents défauts à détecter.	178

GLOSSAIRE

1. **AMCL (Adaptive Monte Carlo Localization)** : méthode probabiliste sur les positions possibles du robot.
2. **ArUco (marqueurs)** : Marqueur carré synthétique composé d'une large bordure noire et d'une matrice binaire interne qui détermine son identifiant.
3. **Air-Cobot** : Pour "Aircraft enhanced Inspection by smaRt Collaborative rObot" ou Inspection améliorée des aéronefs par des robots collaboratifs intelligents.
4. **Algorithme Dijkstra** : Algorithme de trajectoire servant à résoudre le problème du plus court chemin.
5. **Algorithme grille** : méthode d'optimisation qui permet de tester une série de paramètres et de comparer les performances pour en déduire le meilleur paramétrage.
6. **CNN (Convolutionnal Neural Network)** : Les réseaux de neurones convolutifs sont des méthodes d'apprentissage profond qui permettent de traiter les images, inspirés du cortex visuel des animaux.
7. **Calcul de fatigue mécanique** : La fatigue est l'endommagement local d'une pièce sous l'effet d'efforts variables ou de contraintes subies : forces appliquées, vibrations, rafales de vent, etc.
8. **Capteurs extéroceptifs** : Capteurs regroupant les mesures absolues de l'état de l'environnement à fréquence faible (par exemple : GPS, lidar, etc.).
9. **Capteurs proprioceptifs** : Capteurs regroupant les mesures de l'état du robot lui-même (position ou vitesse des roues).
10. **Cinématique directe** : Déterminer une pose et une orientation d'une suite d'articulations en modifiant uniquement les angles de celles-ci et en fonction des longueurs de segments.
11. **Cinématique inverse** : Déterminer une pose d'une suite d'articulations en générant la configuration des joints parents requise et en fonction des longueurs de segments. La solution peut être un ensemble de solutions.

12. **Contrôleur moteur (ou servocommande)** : Amplificateur électronique utilisé pour alimenter et surveiller les rétroactions des servomécanismes.
13. **Contrôle par logique de floue** : La logique floue est une approche utilisée en IA basée sur des "valeurs ou degrés de vérité" sous la forme de chiffres réels compris entre 0 et 1 appliquée aux systèmes de contrôle.
14. **Corrélogramme** : Représente le graphique d'une matrice de corrélation. Une matrice de corrélation est utilisée pour évaluer la dépendance entre plusieurs variables en même temps.
15. **Courbes de Béziérs** : Courbes polynomiales paramétriques développées pour tracer des courbes "à main levée" par l'ordinateur.
16. **Distance focale** : Distance entre le centre optique de l'objectif et le foyer image, où se trouve le capteur de l'appareil.
17. **DoF (Degree of freedom)** : Degré de liberté représentant les possibilités de mouvement d'une articulation.
18. **Déplacement holonome** : Robot se déplaçant sur un plan, disposant de trois degrés de liberté (deux translations et une rotation).
19. **EKF (Extended Kalman Filter)** : Filtre étendu de Kalman. Technique d'estimation applicable aux modèles non linéaires et non gaussiens.
20. **Effets de cage de Faraday** : Enceinte utilisée pour protéger des nuisances électriques et subsidiairement électromagnétiques extérieures ou inversement empêcher un appareillage de diffuser des ondes vers l'extérieur.
21. **Encodeurs de roues** : Systèmes de mesure qui enregistrent les mouvements linéaires au moyen d'une roue et les transforment en valeurs de vitesse ou de position.
22. **Espace confiné** : Volume totalement ou partiellement fermé (bâtiment, ouvrage, équipement, installation. . .) qui n'a pas été conçu pour être occupé de façon permanente par des personnes et au sein duquel l'atmosphère peut présenter des risques pour la santé et la sécurité des personnes.
23. **Fast R-CNN** : Type de réseaux de neurones convolutifs utilisant des régions d'intérêts dans une exécution orientée "temps réel".
24. **Fiducial markers (Marqueurs fiduciaires)** : Objet placé dans le champ de vision d'un système d'imagerie qui apparaît dans l'image produite, pour une utilisation comme point de référence ou comme mesure.

25. **Fins d'effecteurs** : Extrémité d'une suite d'articulation (par exemple : le pied).
26. **GSL (Gas Source Localization)** : Localisation de sources de gaz. Représente l'ensemble des méthodes permettant de localiser la source d'un gaz par un percept chimique.
27. **Gazebo** : Logiciel ROS permettant la mise en simulation de robots.
28. **IBVS (Image-Based Visual Servoing)** : Asservissement visuel basé sur l'image. La caméra est utilisée comme capteur 2D et le mouvement du robot est déterminé pour réduire directement les erreurs entre la position actuelle et les caractéristiques souhaitées dans le plan de l'image 2D.
29. **IMU (Inertial Measure Unit)** : Unité de mesure inertielle. Dispositif électronique qui mesure et rapporte la force spécifique, la vitesse angulaire et parfois l'orientation, en utilisant une combinaison d'accéléromètres, de gyroscopes et parfois de magnétomètres.
30. **Inspection VT** : Inspection "Visual Testing". Première étape d'une inspection utilisant les yeux pour rechercher des erreurs.
31. **Matrice carrée** : Matrice ayant un nombre de lignes et colonnes égales de la forme $(n \times n)$.
32. **Matrice jacobienne** : Matrice des dérivées partielles du premier ordre d'une fonction vectorielle en un point donné.
33. **Mesh (Maille)** : Matériau constitué d'un réseau de fils. Désigne un objet 3D.
34. **Micro-contrôleur** : Circuit intégré compact conçu pour régir une opération spécifique dans un système embarqué.
35. **Milieu contraint** : Environnement limité de son accès dû à sa géométrie ou à une exposition à des risques spécifiques.
36. **Méthode Denavit-Hartenberg** : Permet de caractériser la position relative de deux solides avec seulement quatre paramètres (au lieu de six).
37. **Méthode Levenberg-Marquardt** : Permet d'obtenir une solution numérique au problème de minimisation d'une fonction, souvent non linéaire et dépendant de plusieurs variables.
38. **Méthode Ziegler-Nichols** : Méthode heuristique de réglage d'un régulateur PID.
39. **Norme EN ISO 17637 : 2016** : Définit le traitement du contrôle visuel des soudures par fusion des matériaux métalliques. Elle peut également s'appliquer au contrôle visuel des assemblages avant soudage.

40. **Norme EN ISO 9906-1 : 2012** : Spécifie les essais de performance hydraulique pour l'acceptation par les clients de pompes rotodynamiques (pompes centrifuges, hélico-centrifuges et hélices).
41. **Odométrie** : Technique de mesure des distances parcourues, à l'aide d'un odomètre ou d'un podomètre.
42. **PBVS (Position-Based Visual Servoing)** : Asservissement visuel basé sur la position. La caméra est essentiellement utilisée comme capteur 3D. Il transforme d'abord la mesure de l'image à une erreur dans l'espace euclidien 3D et ajuste le mouvement du robot pour réduire cette erreur.
43. **PID (Primitive Intégrale Dérivation)** : Représente les trois paramètres de contrôle d'un circuit PID. Le but de tout circuit d'asservissement est de maintenir le système à une valeur prédéterminée (point de consigne) pendant de longues périodes.
44. **POI (Points of interest)** : Points d'intérêt. Terme utilisé en cartographie (et donc en référence à des cartes ou des géodatasets) pour le choix de représenter une entité particulière à l'aide d'une icône qui occupe un point particulier.
45. **Percept chimique** : Manifestation d'origine gazeuse d'un élément chimique (Exemple : Fuite de gaz).
46. **Percept visuel** : Manifestation d'un élément dans un spectre visible (Exemple : un défaut de soudure).
47. **Pixel** : Chaque point d'une image électronique.
48. **Protocole XBee** : Protocole de haut niveau permettant la communication d'équipements personnels ou domestiques équipés de petits émetteurs radios à faible portée.
49. **QRcode** : Un type de code-barres à deux dimensions constitué de modules-carrés noirs disposés dans un carré à fond blanc.
50. **R-CNN** : Type de réseaux de neurones convolutifs utilisant des régions d'intérêts.
51. **RGB (Red, Green, Blue)** : Fait référence à un système de représentation des couleurs à utiliser sur un écran d'ordinateur (Rouge, Vert, Bleu).
52. **ROS (Robot Operating System)** : Système d'exploitation pour robot permettant l'intégration d'outils de développement et de communication pour concentrer les tâches de développement sur les applications purement robotiques.

53. **Robot hexapode** : Robot marcheur dont la locomotion est fondée sur trois paires de pattes.
54. **Robot "turtlebot"** : Kit de robot personnel à faible coût avec un logiciel open source ROS. TurtleBot a été créé chez Willow Garage.
55. **SEVESO** : La directive Seveso est le nom générique d'une série de directives européennes qui imposent aux États membres de l'Union européenne d'identifier les sites industriels présentant des risques d'accidents majeurs, appelés « sites Seveso ».
56. **SRI (Système Robotiques d'Inspection)** : Catégorie de robots destinée à l'inspection industrielle et à la reconnaissance de défauts.
57. **SSD** : Type de réseaux de neurones convolutifs qui divise l'image à l'aide d'une grille et fait en sorte que chaque cellule de la grille soit responsable de la détection des objets dans cette région de l'image.
58. **Servomoteurs** : Actionneur rotatif ou un actionneur linéaire qui permet un contrôle précis de la position angulaire ou linéaire, de la vitesse et de l'accélération.
59. **Simulation CFD** : L'utilisation de mathématiques appliquées, de physique et de logiciels de calcul pour visualiser comment un gaz ou un liquide s'écoule - ainsi que comment le gaz ou le liquide affecte les objets lorsqu'il passe.
60. **U2D2** : Convertisseur de communication USB de petite taille qui permet de contrôler et de faire fonctionner DYNAMIXEL avec PC.
61. **YOLO (You Only Look Once)** : Type de réseaux de neurones convolutifs détectant des objets en temps réel. L'algorithme ne nécessite qu'une seule propagation directe à travers un réseau neuronal pour détecter les objets. Cela signifie que la prédiction dans l'ensemble de l'image se fait en une seule exécution d'algorithme.

INTRODUCTION

Le sujet de thèse traite de l'utilisation de la robotique dans une thématique de détection de points d'intérêts significatifs dans le domaine de l'inspection industrielle. Le robot devra détecter des anomalies dans des milieux dégradés et confinés. Dans ce premier chapitre, nous allons, dans un premier temps, évoquer le contexte, les problématiques et objectifs de ce projet de thèse. Nous présentons par la suite l'organisation mise en place pour atteindre les objectifs souhaités. Dans un second temps, nous détaillons l'étude et les critères de cotation des différents éléments contextuels liés à l'environnement autour de la plateforme robotique à développer. Nous terminons d'exprimer les contraintes pour l'expression des besoins de cette étude et une conclusion de fin de chapitre.

1.1 L'inspection robotisée

Depuis plusieurs années, l'inspection industrielle robotisée est devenue un sujet d'étude et d'innovation prenant davantage d'espace dans le domaine de l'industrie. Originellement, l'inspection robotisée a été développée pour contrôler des équipements aéronautiques, figure 1.1 [1], pour le domaine nucléaire et la gestion de ces infrastructures.



FIGURE 1.1 – Air-Cobot (Aircraft enhanced Inspection by smaRt Collaborative rOBOT), a été développé pour contrôler la qualité des différentes pièces aéronautiques par des contrôles non-destructifs [1].

L'inspection industrielle peut revêtir une importance capitale pour le maintien des équipements et installations spécifiques et la sécurité des utilisateurs, notamment dans le domaine des énergies et des transports. En effet, l'inspection et le maintien d'équipements industriels sont une tâche critique sur plusieurs points. En premier lieu, il faut mobiliser des ressources humaines techniques de manière occurrente sur les installations à inspecter, avec une formation appropriée. À cela, on peut ajouter des endroits difficiles d'accès ou avec des contraintes environnementales fortes pour les opérateurs (atmosphères explosives, nucléaires, milieux hyperbares, etc.). De plus, les équipements industriels de grandes envergures (les centrales, raffineries, etc.); entreprises et sous-traitants doivent se coordonner pour organiser les différentes actions d'inspections et maintenance. Enfin, ce type d'intervention a un coût et est sujet à la potentielle erreur humaine ou à une instabilité de l'environnement pouvant engendrer des catastrophes industrielles importantes.

Au vu des différentes contraintes que cela implique, appréhender l'environnement industriel est un point important de ce type d'application. La progression des systèmes embarqués, de la vision par ordinateurs, des algorithmes d'apprentissage, l'utilisation de robots pour les tâches d'inspection s'est peu à peu légitimé dans le monde industriel. Les systèmes robotiques d'inspection, SRI, sont généralement conçus pour les applications intérieures telles que les sites industriels. Ces technologies à coût raisonnable réduisent considérablement les frais d'acquisition et de maintenance [26]. Les opérations sont alors facilitées et peuvent être utilisées dans de plus larges applications de surveillance.

Néanmoins, des environnements complexes restent difficiles à appréhender en raison de plusieurs caractéristiques. La géométrie, la situation sur site, les contraintes photométriques, l'occurrence et le temps d'inspection, la dangerosité, le caractère évolutif et incertain de l'environnement compliquent la réalisation de l'inspection de manière automatique. L'anticipation de tous les cas de figure serait un axe de recherche à part entière. Parmi ces environnements, nous pouvons notamment citer les cuves industrielles dans le secteur pétrolier et gazier, l'inspection interne des navires, les carrières souterraines, ou encore les environnements post-catastrophes.

1.1.1 Contexte de la thèse

Dans le cadre de ce travail de recherche, des cas d'applications ont été définis avec l'entreprise d'accueil. Dans cette thèse, encadrée par l'ENSTA Bretagne, CESI LINEACT et Pilgrim Technology, figure 1.2, sous la modalité d'une thèse CIFRE, nous aborderons l'utilisation de robots pour des tâches d'inspection dans le domaine de l'industrie et des environnements complexes.

Cette thèse a été réalisée avec Pilgrim Technology, une entreprise d'inspection industrielle réalisant des contrôles pédestres et par drone. Elle construit ses drones sur mesure en fonction de l'opération à réaliser et intègre des capteurs. Suivant le rendu demandé par le client, l'entreprise adapte le drone avec les capteurs nécessaires à l'acquisition en temps réel des données et au traitement hors ligne une fois l'inspection terminée. Cette stratégie s'organise autour des besoins des différents industriels en termes d'inspection et d'acquisition des données souhaitées. Principalement orientée sur les domaines pétroliers et gaziers, énergie renouvelable, nucléaire, marine, l'entreprise s'est d'abord focalisée sur la conception de drones et le traitement des données adapté aux différents besoins client [27].

Pour Pilgrim Technology, la robotique est une innovation qui doit permettre :

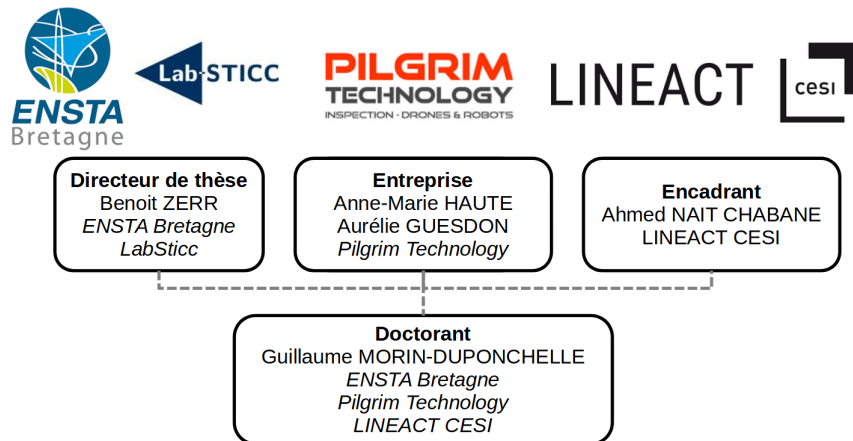


FIGURE 1.2 – Organisation de l'équipe encadrant la thèse.

- Améliorer la productivité et qualité des inspections en s'affranchissant des limites liées aux capacités humaines.
- Réduire la pénibilité des inspections (physique, psychologique, cognitive, contraintes liées à l'environnement).
- Réduire les coûts du client final sur le maintien des équipements.
- Apporter des données clé aux clients permettant d'améliorer, corriger des éléments qu'il n'aurait pas pu avoir avec une inspection pédestre.
- Éviter les accidents ou incidents liés à des défauts dans l'installation.

Souhaitant traiter les cas difficiles d'inspection terrestre pour les installations pétrolières, géologiques et marines, une thèse a été proposée par Pilgrim Technology en robotique et intelligence artificielle.

La thèse CIFRE est réalisée en partenariat avec le CESI LINEACT et l'ENSTA Bretagne afin de s'appuyer sur des compétences en robotique et intelligence artificielle dans l'industrie.

L'ENSTA Bretagne est une école d'ingénieur et doctorale possédant une expertise en hydrographie/océanographie, systèmes d'observation/IA/système embarqués, robotique autonome, systèmes numériques et sécurité, et architecture navale, et offshore. Benoit ZERR, Enseignant chercheur à l'ENSTA Bretagne a été mon directeur de thèse. Ses recherches ont porté sur les capteurs pour les fonds marins et la robotique marine. Le CESI LINEACT est un laboratoire de recherche et d'innovation avec comme cœur de sujet l'industrie 4.0. L'objectif de ce laboratoire est d'accompagner les entreprises, partenaires à relever les différents défis technologiques liés à l'association du numérique, de la robotique

et cobotique dans l'industrie. Ahmed NAIT CHABANE, Enseignant chercheur au CESI Lineact a été mon co-encadrant de thèse. Ses recherches portent sur le traitement du signal et d'images, l'apprentissage automatique appliqué à l'industrie 4.0 et les algorithmes de décisions.

1.1.2 Problématiques et objectifs de la thèse

Depuis plusieurs années, les solutions apportées pour améliorer la productivité, la précision et l'automatisation des procédés d'inspections sont fournies par la robotisation. Cette tendance, observée aussi dans le domaine de la production industrielle, vise à faire collaborer l'humain avec des ordinateurs et robots afin de produire des inspections moins coûteuses, augmenter la quantité de données récoltées et traitées et ainsi éliminer les variabilités et risques d'erreurs humaines dans des environnements toujours plus complexes et potentiellement dynamiques.

Cette philosophie, centrée sur la technologie au service de l'humain, s'est diffusée dans le domaine de l'inspection, dynamisée par les besoins évolutifs et techniques des industries, notamment sur les entreprises classées SEVESO [2]. Les entreprises concernées par la directive européenne SEVESO sont définies pour leurs activités liées à des substances dangereuses (Par exemple : les dépôts pétroliers ou explosifs, les raffineries, les sites pétroliers et chimiques.). Cependant, l'inspection reste une tâche délicate tant sur le plan visuel, cognitif et locomoteur [28].

La conception de tels robots implique une architecture plus complexe avec d'avantage de capteurs, d'actionneurs, de contrôleurs. L'identification des défauts visuels nécessite un traitement d'images permettant de discriminer le point d'intérêt souhaité. Le calcul des fonctions de coût lié à la carte est des enjeux majeurs pour l'obtention d'une navigation adaptée aux espaces contraints. Les stratégies de décision, d'adaptation, d'apprentissage peuvent être des calculs anticipés, à travers un modèle ou un logigramme. L'accès à ses défauts peut nécessiter une locomotion de robot spécifique, permettant par exemple, de franchir des obstacles, de pouvoir évoluer sur un sol glissant ou nivelé, ou tout simplement de pouvoir observer l'ensemble d'un volume.

Mais cette évolution comporte plusieurs effets pour les organismes de conception de drones et robots personnalisés :

- Le coût d'investissement, de programmation et de maintenance est supérieur à un produit polyvalent.

- La maîtrise du procédé de fabrication (qualité, coût, délai), de conception, et de réalisation est difficile à appréhender et entraîne un coût important. Par exemple, l'importation de pièces en carbone usinées ou d'électronique de contrôle ou calculatoire généralement expédié par la Chine.
- La maîtrise des différentes situations possibles dans une inspection est compliquée à prévoir.
- Les problèmes de communication en mode infrastructure entre la station de commande et le robot dans des environnements spécifiques (de type cage de Faraday, peinture et tissus blindés, couloir long avec des embranchements, milieu confiné, ligne haute tension, lignes téléphoniques, émetteurs ou stations radio, clôtures électriques, radars, relais de diffusion de télévision, etc.) entraînent des difficultés voire incapacité de pouvoir télé opérer le robot.
- Les milieux dangereux (chimique, nucléaire, radioactif, instabilité mécanique, etc.) restreignent l'utilisation de procédés de téléopérations et le téléopérateur.
- Les environnements contenant de la poussière ou du gaz rendent difficile l'utilisation de drones en raison de la qualité dégradé du flux vidéo final et des risques de perte de liaison radio.
- L'inspection de fuites de gaz dans les milieux confinés nécessite une configuration de capteurs spécifiques qui correspond au gaz recherché. Pour un robot terrestre, le gaz doit être plus lourd que l'air pour atteindre les capteurs.

En outre, la robotisation systématique des procédés d'inspection suggère que l'humain disparaît de l'équation. Toutefois, comme pour la production robotisée industrielle, le téléopérateur ou l'inspecteur garde une place centrale dans le processus d'inspection. En effet, la supervision de la mission, programmation, maintenance, l'expérience technique, l'intervention humaine reste nécessaire. Cette utilisation ouvre aussi le champ de possibles en matière d'innovation et démontre une solution pour appréhender les environnements contraints et hostiles. Elle offre aussi une mutation dans la construction des équipes de compétences et savoir-faire techniques en lien avec les procédés et à la technologie employés. Où l'expérience métier est directement transférée sur une unité autonome. Par exemple, les procédés d'apprentissage pour la détection de défauts requièrent une expertise métier dans l'étape de labellisation ou validation des bases de données d'apprentissage.

Fort de ce constat, le développement d'un robot conduit naturellement à l'étude de l'environnement d'évolution et aux contraintes que cela implique. Étant donné que les té-

léopérateurs et inspecteurs sont les experts de leur domaine respectif et actuels, il semble évident de prévoir un système qui évite de présenter un nouveau risque pour les utilisateurs par son fonctionnement ou possible dysfonctionnement. Il doit répondre aussi aux contraintes de l'environnement et à l'impact du milieu sur le système de captation ou localisation. Le robot sera amené à évoluer dans un environnement spécifique où il devra inspecter les anomalies chimiques (concentration de gaz) et visuelles (défauts de soudure, fissures, corrosion).

Le sujet de recherche d'analyser et d'inspecter automatiquement les percepts visuels ou chimiques dans des environnements contraints et dégradés, entraîne diverses problématiques.

Le premier axe de recherche est de définir une architecture robotique, des capteurs et un mode d'inspection permettant d'inspecter l'environnement en question. De plus, l'utilisation d'intelligence artificielle pour la loi de commande ou pour l'aspect visuel, définit les résultats de performances en fonction de l'environnement. À la connaissance de l'auteur, il n'existe actuellement pas d'outils de décision permettant de définir un robot, les capteurs en fonction d'un type d'inspection ou d'un type d'environnement.

L'inspection industrielle est une activité complexe qui associe des interactions visuelles et motrices dans un environnement spécifique. La deuxième problématique entraîne une réflexion sur la définition des critères d'étude de l'environnement pour traiter les différents cas d'inspection.

Le troisième questionnement est de définir le système de contrôle et les stratégies à adopter pour la localisation, la cartographie, le franchissement d'obstacles en fonction du système choisi. Cet aspect est traité par les données de sorties de l'analyse de l'environnement.

Le quatrième sujet à caution est de définir l'architecture matérielle pour les effecteurs, les microcontrôleurs et systèmes embarqués pour achever les asservissements dans un objectif de fonctionnement en temps réel.

La cinquième (et dernière partie) du sujet de recherche est de pouvoir qualifier la solution robotique et les choix algorithmiques réalisés.

Les enjeux et le traitement de ses différentes parties permettront d'établir une méthode d'analyse sur des cas spécifiques de robotique mobile et une méthode d'asservissement visuel et chimique. Pour répondre à ces différents enjeux, les objectifs de la thèse sont multiples. Le premier objectif sera d'analyser les différents cas d'application et leurs points d'intérêts. Il faudra aussi définir les éléments contextuels de la recherche et analyser l'environnement. Les points d'intérêts à détecter et les caractéristiques de l'environnement

auront une forte influence sur le choix du matériel. La plateforme robotique devra répondre aux contraintes de l’environnement. La partie algorithmique concernant la locomotion et la détection des points d’intérêts visuels et chimiques permettront l’inspection automatisée. La performance du système sera à évaluer en fonction des cas d’application en simulation ou en réel. Les critères de performances sont définis par le temps d’inspection, la précision du positionnement du robot par rapport à l’objet et la localisation du robot par rapport à son emplacement dans son espace. Nous développons dans cette thèse une démarche de conception d’un robot et son application dans différents scénarios possibles selon une perspective d’inspection industrielle et de recherche d’anomalies. Le but est d’inspecter des sites dangereux ou inaccessibles pour les humains avec une détection et reconnaissance des différents points clé à vérifier lors de l’inspection.

Pour cela, 3 cas d’application ont été identifiés de difficultés croissantes :

- Le premier cas concerne les inspections des cuves 1.3 et espace de stockage 1.4 dans le domaine pétrolier et gazier en espace confiné. Nous associons aussi, le milieu naval confiné comme les couloirs de bateau avec une problématique de franchissement d’obstacles. Dans ce cas, la détection des fissures, de la corrosion et des concentrations de gaz (hydrocarbures) permettent de prévenir des risques de points chauds et présence de fuites de gaz.



FIGURE 1.3 – Sphère de stockage de gaz butane à Abidjan.

- Le deuxième cas concerne les mines et les carrières souterraines figure 1.5 (A). Autrefois exploitées pour permettre, l’extraction de matériaux de construction (calcaire, argiles, craie, etc.), ces cavités, creusées dans le sous-sol, n’ont par la suite pas été remblayées systématiquement pour des raisons financières [5]. Ces carrières, laissées telles qu’elles, subissent des affaissements ou des effondrements. Afin de prévenir ces risques, les points



FIGURE 1.4 – Citerne de stockage à inspecter.

d'intérêts notables sont la présence d'eau, de fissures et d'éboulis.



FIGURE 1.5 – A : Carrière souterraine de calcaire (60 hectares) à Conflans Sainte Honore, Herblay [5]. B : Carrière et effondrement d'une maison à Casablanca suite à de forte précipitation [30].

- Le troisième cas consiste à réaliser l'inspection sur des sites concrets de sinistre post-catastrophe dans un milieu urbain, figure 1.5 (B), de type séisme et éboulement, où l'environnement a subi une déformation importante et donc une géométrie des lieux d'inspection modifiée et chaotique [30]. Dans ce cas critique, la détection de gaz domestique (usuellement butane ou propane) et la détection de signatures thermiques (humains, animaux) peuvent s'avérer des points d'intérêts cruciaux pour éviter le sur-accident et gagner du temps sur le sauvetage.

Le sujet de thèse porte donc sur l'inspection automatique sur ces différents cas d'application et aussi en lien avec des environnements similaires sur l'inspection de l'intérieur de bateau, par exemple. L'analyse des différents environnements est donc primordiale pour définir une plateforme robotique adéquate. Dans le cadre d'inspection de cuves industrielles, de mines ou carrières souterraines, le drone aérien devient inadapté pour plusieurs raisons. Les drones étant généralement télé-opérés par communication, radio-fréquences, les environnements de faibles volumes avec de fortes discontinuités entraînent une perte du signal et du contrôle du drone inévitable [31]. De plus, l'effet des moteurs du drone sur

le sol génère un nuage de poussières [32] qui diminue significativement le rapport signal à bruit des images et vidéos acquises et augmente le risque d'endommager le drone ou de dégrader l'environnement lors de la navigation. L'intérêt de l'utilisation d'un robot est de pouvoir automatiquement reconnaître les anomalies visuelles, de se diriger de manière autonome vers celle-ci et ainsi les marquer sur une cartographie du milieu inspecté.

Dans ce travail de recherche, nous nous focalisons sur les points d'intérêts visuels (défauts surfaciques) et chimiques (émanation de gaz). L'objectif scientifique de ce projet de recherche est l'étude de la reconnaissance automatique de points d'intérêts (POI) pour un robot d'inspection dans un environnement contraint et dégradé. Les POI détectés par le robot seront exploités pour définir un itinéraire d'exploration vers l'origine du percept. Cette tâche intervient dans une démarche d'évaluation de l'environnement (inspection) et permet de définir les risques avant une exploration par un être humain. Cette reconnaissance nous permettra d'adapter le comportement du robot en fonction des points d'intérêts détectés précédemment et d'optimiser le trajet du robot.

1.1.3 Organisation de la thèse

La deuxième partie du chapitre 1 vise à analyser les différents éléments et contraintes constituant les 3 cas d'application. Dans cette partie, nous utilisons des critères définis préalablement. D'autres critères ont été ajoutés en lien avec les environnements d'applications et avec le métier d'inspection d'industrielle. Cette évaluation permettra, par la suite, de définir les axes de recherche, l'architecture de captation et ainsi anticiper les difficultés potentielles liées aux environnements et à l'appréciation. Le chapitre 2 aborde les différents état de l'art en lien avec le sujet de thèse : la reconnaissance automatique de points d'intérêts pour un robot d'inspection dans un environnement contraint et dégradé. Dans un premier temps, nous nous intéressons aux différentes plateformes de la robotique terrestre en établissant un système de cotation basé sur les travaux de Aravind Seeni [115] en prenant en compte aussi la manœuvrabilité et la tolérance à la faute sur la partie locomotion. Également, nous aborderons plus en détail l'architecture robotique la plus adaptée, suite à ce travail d'analyse. Enfin, dans cette dernière partie, nous examinerons les différentes stratégies possibles pour effectuer la détection d'objets, de gaz et enfin les stratégies pouvant être adoptés pour réaliser des asservissements visuels et chimiques.

Dans le chapitre 3, nous proposons une méthode pour effectuer la cinématique du robot et ainsi produire un contrôleur prenant en compte la géométrie et l'architecture du robot.

Nous proposons aussi une méthode de localisation et de navigation permettant, in fine, la cartographie et la planification de trajectoire. Des méthodes existantes d’asservissement visuels et chimiques sont mises en avant. Nous proposons 5 méthodes d’asservissement chimique qui seront évaluées dans le dernier chapitre. L’approche théorique d’asservissement visuel inclut une méthode de détection d’objets basée sur les réseaux de neurones convolutifs et un calcul de Jacobien pour la cinématique inverse. Nous proposons une approche pratique avec un cas d’application simple d’asservissement visuel en utilisant un QRcode.

Le chapitre 4 révèle les différentes itérations d’implémentation et d’intégration du robot développé dans ce travail. Différentes versions du robot sont mises en avant et témoignent de l’évolution de l’architecture. On explique, par ailleurs, l’intégration des différentes briques technologiques permettant de réaliser la navigation, l’asservissement visuel et chimique. Dans cette partie, nous proposons d’intégrer l’ensemble des briques logicielles dans l’architecture ROS (Robot Operating System) sous Linux. Nous utilisons Gazebo pour effectuer les essais en simulation.

Les expérimentations et résultats sont présentés dans le chapitre 5. Dans la première partie de chapitre, nous évaluons les performances odométriques, de navigation, d’asservissement visuel et chimique dans un milieu contraint dans un environnement de simulation. Dans la seconde partie, nous aborderons les essais avec le robot réel, dont des essais de statique et de dynamique afin d’évaluer les capacités motrices. Puis, nous aborderons les résultats d’une démonstration réalisée dans une course de bateau, pour évaluer les capacités de franchissement du prototype.

1.2 Étude des éléments contextuels

Dans cette partie, nous allons aborder les thématiques encadrants ce sujet de recherche. Ce travail préparatoire permet une vision globale de l’ensemble des caractéristiques environnementales lié à la détection des POI prévues dans ce travail de thèse. L’objectif est d’obtenir un support de réflexion permettant de définir les différents éléments de l’architecture physique et logicielle du robot d’inspection. De plus, cela permettra de s’approcher d’une résolution des trois cas d’étude. Il est déterminant de se baser sur les caractéristiques environnementales afin de tenter de résoudre les problématiques de cette thèse avec un outil matériel unique où seuls les algorithmes de détection et éventuellement les capteurs changeront en fonction des POI à détecter. Ici, les éléments contextuels considérés

comme les plus complexes sont une mission longue (plusieurs batteries nécessaires), dans un environnement intérieur et irrégulier (obstacles, possibilité de mouvements du sol...), sans lumière, à dimensions volumiques inconnues. Bien entendu, il sera indispensable au fur et à mesure du déroulé de cette recherche, de déterminer les limites opérationnelles du robot développé. Néanmoins, ce premier état offre un axe de réflexion en ce qui concerne la nécessité de développer un robot polyvalent en termes de navigation, d'évitement d'obstacles, ainsi que du besoin d'emport de charge (lumière, capteurs, station de calcul, etc.) important même si encore à définir.

1.2.1 Inspection des milieux contraints

L'appréhension des environnements complexes a été un enjeu fort durant ces dernières années. Les possibilités fonctionnelles et la sécurisation des environnements complexes qu'offrent les systèmes robotiques, permettent d'appréhender au mieux ces environnements. Évidemment, il convient de prendre en compte les différentes caractéristiques qui les composent. Dans la cadre de cette thèse, nous rappelons que nous nous intéressons à 3 cas d'études de complexité croissante :

- Le cas 1, l'inspection est réalisée dans un espace confiné (milieu industriel, pétrolier, navires, domaine des énergies).
- Le cas 2, l'inspection est réalisée dans un environnement avec un risque avéré de perte de signal et à dominante rocheuse (mine, carrière souterraine).
- Le cas 3, l'inspection est réalisée dans un environnement ayant subi des déformations importantes et donc une géométrie des lieux d'inspection modifiée et chaotique (catastrophes naturelles de type séisme ou éboulement dans un environnement urbain).

1.2.2 Les milieux contraints et dégradés

Par définition, les milieux contraints et dégradés se réfèrent à deux appréhensions de l'environnement. Les milieux contraints sont définis comme des espaces avec une ou plusieurs restrictions de mouvements impliquant des conditions difficiles d'accès [3]. Les milieux dégradés impliquent une notion de modification spatiale tant dans la géométrie ou dans le volume, à une forme de détérioration pouvant entraîner des risques (industrielles, humains, structurelles, etc.) [4]. Ainsi, les 3 cas d'études à traiter dans cette thèse comportent ces deux dimensions : contraints et dégradés. Ces cas sont le théâtre d'une activité humaine nécessaire pour maintenir en état et évaluer l'état structurel, repérer les

anomalies et effectuer les correctifs nécessaires à son maintien. C'est pourquoi les infrastructures de gaz, les carrières souterraines et l'environnement urbain post catastrophe sont des milieux considérés comme contraints et dégradés. Les moyens humains à déployer peuvent être importants pour explorer ce type de zone et l'approche robotique semble être une solution adaptée dans ce type de situation.

Le cas 1 : Le stockage d'hydrocarbure et gaz et les milieux navals

Dans le domaine du stockage de liquide et de gaz, les constructions métalliques sont majoritaires. La maintenance périodique des dispositifs de stockage de pétroles et gaz [101] comme ceux des navires [102] permettent de détecter des défauts dans la structure métallique qui les compose par différentes techniques visuelles : seuillage d'histogramme, segmentation des images et la morphologie mathématique [101]. Parmi ces défauts, on retrouve la dégradation, les fissures et la corrosion du revêtement. Ces défauts sont des indicateurs importants de l'état de la surface métallique [103] et peuvent prévenir les départs de feu ou fractures, et, in fine, éviter les catastrophes aux conséquences humaines, environnementales et financières que cela peut engendrer [102]. Pour éviter d'atteindre de telles situations indésirables, les inspections à bord les navires de mer sont des activités régulières lancées en partie en raison de la classification applicable et réglementations statutaires, et en partie en raison de l'intérêt évident des exploitants de navires et des armateurs à anticiper les situations défectueuses, pour des raisons de sécurité, mais aussi en raison des coûts associés à des interruptions inattendues impactant la disponibilité du service des navires. Ces inspections se font généralement à des coûts importants dus à l'emploi d'inspecteurs formés à ce type de défauts, ainsi que le besoin de maintenir un accès sécurisé. Ainsi, les défauts sont principalement visuels et concernent la surface : détection de corrosions, de fissures, d'irrégularités.

Le cas 2 : L'exploration et l'inspection de mines et carrières souterraines

L'exploitation des mines est une activité très ancienne, certains gisements, notamment le fer (figure 1.6), était en forte abondance et ne nécessitait pas par forcément d'outils sophistiqués, car facilement accessibles. La croissance de l'industrie métallurgique, automobile et informatique nécessite toujours plus de matière première. Cette exploitation a longtemps été considérée comme périlleuse et a été très souvent effectuée de manière manuelle pour l'extraction de matières premières à l'aide de système de convoyeurs pour le déplacement.



FIGURE 1.6 – Mine de fer en Lorraine [104].

L'inspection de tels environnements [105] comporte plusieurs risques. Tout d'abord, la localisation des dépôts, la géométrie (couches fines), l'environnement difficile (poussière, température et humidité) y compris les risques naturels (émission de gaz, eau et événements sismiques). Face à ces conditions, la demande de solutions robotiques (provenant d'industriels, d'entreprise, de chercheurs) reste un enjeu encore d'actualité. Même si de nombreux exemples et preuves de concepts comme des systèmes autonomes, robots, ou encore drones ont vu le jour, l'introduction de robots dans les mines reste limitée [106] - [108]. Les critères retenus sont donc la géométrie des lieux, les particules en suspension, la température et les émissions de gaz.

Le cas 3 : L'exploration d'infrastructures urbaines post-accident

Beaucoup de zones dans le monde sont affectées par des calamités naturelles. Ces désastres sont des événements exceptionnels avec un coût important (financier, humain et logistique) pouvant être d'origine humaine ou naturelle [109] comme les attaques terroristes, les tremblements, les incendies, les inondations, etc. Ces événements créent des situations d'urgence nécessitant une organisation rapide et coordonnée pour secourir les victimes. Les causes d'échecs de sauvetage sont principalement dû au fait que les personnes restent prisonnières de ces désastres, ou que le délai d'extraction des victimes est trop long.

Les procédures conventionnelles de recherche et sauvetage se font habituellement en utilisant des moyens humains et un système de prise de décision incertaine. Cet axe de recherche est historiquement considéré sous l'angle des sciences physiques (volcanologie,



FIGURE 1.7 – Image aérienne d’un séisme de magnitude 6,8 en Turquie, dans une zone urbaine [110].

sismologie, hydrologie, géomorphologie). Mais pour réduire les risques (pour les victimes et les sauveteurs), une compréhension de la situation, une stratégie de prise de décision et un développement technique est nécessaire.

Synthèse

La robotique peut donc jouer un rôle important dans le délai de détection et d’extraction des personnes impactées par des sinistres. Néanmoins, l’appréhension de l’environnement reste incertaine, car la géométrie reste aléatoire et risque d’avoir un comportement dynamique et ainsi de modifier l’environnement (effondrement par exemple). Les critères à prendre en compte dans ces types d’environnement peuvent être nombreux et incertains, dépendant de l’événement, de la géométrie des lieux, de la qualité du sol, de la densité de population et des infrastructures, etc. Dans notre cas, nous allons considérer le thème des destructions partielles et mécaniques d’infrastructure, pouvant avoir une origine naturelle (séisme, vétusté, malfaçons, etc.), figure 1.7, [110] ou d’origine humaine.

Nous avons vu précédemment que les 3 cas d’application sont considérés comme des environnements contraints et dégradés. Afin d’apporter une orientation plus précise, le postulat de l’étude est d’évaluer les environnements en question. Dans la partie suivante, nous allons proposer des critères d’évaluation pour guider les choix technologiques et apporter la réponse la plus pertinente aux 3 cas d’application.

1.2.3 Les critères d'évaluation

L'étude d'un environnement a été longtemps un axe de recherche majeur tant dans les sciences du vivant (biologie), de ce qui a vécu (paléontologie, archéologie, etc.), de ce que l'Humain a exploré partiellement (océanographie, géologie) de ce que l'on a peu exploré (astrophysique). L'étude d'un environnement incluant les interactions avec un robot doit être traitée de manière scientifique, afin d'appréhender au mieux le choix du robot et ses capacités nécessaires pour l'élaboration et la réussite de la tâche demandée. L'évaluation des cas d'étude apporte une source d'entrées importantes pour définir les principales contraintes d'une opération, du robot, des capteurs, des algorithmes et de la stratégie de recherche et développement à adopter pour le sujet. Dans les paragraphes suivants, nous aborderons un choix de critères basé sur l'environnement, l'inspection à réaliser, les types de percepts (visuels et chimiques), et enfin, nous ajouterons la cotation du risque pour prendre en compte les milieux dangereux pour les robots et les humains, bien évidemment. Un outil de cotation d'une inspection robotisée peut permettre de discriminer et d'évaluer la complexité globale de cette tâche. Les critères sont différenciés en plusieurs catégories. L'environnement de la scène, traitera des caractéristiques liées aux paramètres photométriques et géométriques, ainsi qu'à l'évolution possible. Le type d'inspection révélera les critères à identifier, la récurrence de l'inspection et les types de percepts nécessaires.

Les critères sont répartis en 5 catégories : (1) les critères liés à l'environnement présenté dans le tableau 1.1, (2) les critères liés à l'inspection dans le tableau 1.2, (3) les critères liés aux risques dans le tableau 1.3. Enfin (4) les critères liés aux percepts visuels et (5) chimiques sont présentés respectivement par les tableaux 1.4 et 1.5. Les critères visuels (4) et chimiques (5) sont en relation directe avec les besoins de l'entreprise et aussi des POI exploités dans cette thèse. Ces différents critères peuvent être enrichis par d'autres en fonction du type d'inspection à réaliser, du POI à rechercher ou encore de l'environnement.

Critères liés à l'environnement

La première catégorie est l'environnement, tableau 1.1. Dans ce tableau, les informations concernant les contraintes d'accessibilité et géométriques sont définies par 9 critères (de 1 à 9). Ces critères sont les suivants : (1) le type d'environnement (2) les matériaux de l'environnement pouvant impacter les capteurs du robot (par exemple, les environnements métalliques entraînant des effets de cages de Faraday), (3) la géométrie des lieux (4), la surface et (5) la hauteur de l'espace à inspecter. Les critères (6), (7) et (8) permettent de définir la stratégie d'évolution du robot dans l'environnement avec les critères (6) pour les obstacles (7) la dynamique de l'environnement (activité humaine, stabilité du milieu) et enfin (8) la trajectoire du robot à réaliser. Le critère (9) cotation d'environnement somme l'ensemble des critères évoqués précédemment pour évaluer la difficulté de l'environnement et la stratégie à adopter d'un point de vue recherche et développement (simulation de l'environnement) et capacités nécessaires de franchissement pour le robot.

Tableau des critères d'évaluation de l'environnement								
Catégorie	N°	Critères	Niveau 1		Niveau 2		Niveau 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Environnement	1	Type d'environnement	1	Environnement extérieur	3	Environnement extérieur et intérieur	9	Environnement intérieur exclusivement
	2	Matériaux de l'environnement / Capteurs du robot	1	Matériaux avec peu d'incidences sur les capteurs	3	Matériaux avec incidences moyennes sur les capteurs du robot	9	Matériaux avec incidences photométriques (bruit), localisation (effet cage de Faraday), ou autres perturbations importantes. Risques de perturbations capteurs avérés.
	3	Géométrie des lieux	1	Géométrie simple (forme élémentaire).	3	Géométrie élaborée (assemblage d'éléments simples).	9	Géométrie complexe ou déstructurée.
	4	Surface	1	Inférieure ou égale à 250 m ² .	3	Entre 250m ² et 500m ²	9	Supérieure à 500 m ²
	5	Hauteur	1	Inférieure ou égale 3 mètres.	3	Entre 3 et 5 mètres.	9	Supérieure à 5 mètres
	6	Obstacles	1	Pas d'obstacle.	3	Moins de 5 obstacles de moins de 10 cm.	9	Supérieure à 5 obstacles de 10 cm et plus.
	7	Dynamique de l'environnement	1	Inerte.	3	Faibles activités humaines et/ou environnement partiellement stable.	9	Activité humaine forte et/ou environnement instable avéré.
	8	Trajectoire nécessaire	1	Navigation en ligne droite possible.	3	Plusieurs points de navigation avec peu de contraintes de trajectoire.	9	Une majorité de points avec un besoin d'anticiper l'itinéraire. Trajectoire contrainte.
	9	Cotation d'environnement	$\Sigma < 24$	Environnement à faibles difficultés	$24 < \Sigma < 48$	Environnement à appréhender et robot à adapter.	$48 < \Sigma < 72$	Environnement complexe à fortes contraintes. Test en simulation.

TABLE 1.1 – Tableau des critères contextuels d'environnement

Critères liés à l'inspection

La deuxième catégorie est l'inspection, tableau 1.2. Ici, les informations concernent les contraintes de l'inspection à réaliser et sont définies par 5 critères (de 10 à 14). Ces critères sont les suivants : (10) les types de points d'intérêts (visuels, chimiques, thermiques, etc.), (11) le niveau technologique pour la détection des POI, (12) la durée de l'inspection (en lien avec l'autonomie nécessaire du robot) et (13) la récurrence (certaines inspections réglementaires nécessitent une périodicité.). Enfin, le critère (14) évalue la difficulté de l'inspection et la technicité de l'inspection. Dans cette partie, on souligne le fait d'adapter les capteurs et les algorithmes aux points d'intérêts, ainsi que l'autonomie du robot nécessaire.

Tableau des critères d'évaluation de l'inspection								
Catégorie	N°	Critères	Niveau 1		Niveau 2		Niveau 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Type d'inspections	10	Types de points d'intérêts	1	1 type de point d'intérêt. Visuel ou volumétrique.	3	2 types de points d'intérêt. Visuel et chimique par exemple.	9	3 types de points d'intérêts et plus. Visuel, chimique, volumétrique ou autre point d'intérêt d'une autre nature.
	11	Niveau technologique	1	Nécessite des capteurs peu coûteux (inférieur à 50€ unité) disponible sur le marché.	3	Nécessite des capteurs de bonne précision (entre 50 et 200€) avec des librairies en source libre.	9	Nécessite des capteurs spécifiques ou de l'industrie (supérieur à 200€) avec un kit de développement logiciel constructeur.
	12	Durée	1	Inspection rapide. Moins de 30 minutes.	3	Inspection moyenne. Entre 30 minutes et 1 heure.	9	Inspection longue. Supérieure à 1 heure.
	13	Récurrence	1	Nécessite une seule visite pour inspecter sur un événement précis.	3	Nécessite des visites épisodiques (4 visites sur une année) pour un environnement ou une installation à surveillance modérée.	9	Nécessite des visites récurrentes (supérieure à 4 visites par mois) pour un environnement ou installation avec une surveillance renforcée.
	14	Cotation d'inspection	$\Sigma < 12$	Niveau d'inspection faible. Accessible à la majorité du personnel. Sollicitation faible.	$12 < \Sigma < 24$	Niveau d'inspection moyen. Nécessite du personnel formé. Sollicitation moyenne.	$24 < \Sigma < 36$	Niveau d'inspection fort. Nécessite du personnel qualifié. Forte sollicitation.

TABLE 1.2 – Tableau des critères contextuels d'inspection.

Critères liés au risque

La troisième catégorie est le risque encouru par le robot, tableau 1.3. Comme l'inspection robotisée permet de repousser les limites d'exploration des environnements contraints, dégradés et potentiellement dangereux ; une cotation du risque est aussi appliquée. Cette thématique permet de définir si des stratégies ou comportements de récupération sont à prévoir en fonction des dommages infligés sur le robot. Dans ce tableau, les informations concernant la cotation du risque par rapport à nos différents cas d'études sont définis par 3 critères (de 16 à 18). Les critères sont les suivants : (16) la fréquence d'exposition donnée par le critère durée (12) et récurrence (13) dans le critère d'inspection), (17) la

gravité (le degré de dommage possible sur le robot), (18) la maîtrise du risque. Enfin, la cotation finale du risque (19) somme les critères évoqués précédemment. Cette évaluation vise aussi à déterminer les composants structurels, la potentielle redondance électronique et les éléments à renforcer.

Tableau des critères d'évaluation du risque								
Catégorie	N°	Critères	Niveau 1		Niveau 2		Niveau 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Évaluation du risque	16	Fréquence d'exposition Durée (n°12) + Récurrence (n°13)	$\Sigma < 6$	Faible exposition.	$6 < \Sigma < 12$	Exposition moyenne.	$12 < \Sigma < 18$	Exposition fort.
	17	Gravité	1	Faible. Dégâts esthétiques ou superficiels.	3	Moyenne. Dégâts légers mécaniques ou réparable. Empêche temporairement la progression.	9	Fort. Dégâts lourds sur le robot empêchant sa progression de manière définitive.
	18	Maîtrise du risque	1	Forte. Présence d'éléments de maîtrise prioritaire et secondaire assurant un haut niveau de prévention et de protection.	3	Moyenne. Présence d'éléments de maîtrise secondaire.	9	Absence de mesure ou mesure inadaptée.
	19	Cotation du risque	$\Sigma < 12$	Risque faible.	$12 < \Sigma < 24$	Risque moyen.	$24 < \Sigma < 36$	Risque fort.

TABLE 1.3 – Tableau d'évaluation des critères contextuels du risque.

Critères liés aux percepts visuels et chimiques

La quatrième et cinquième catégorie concernent les percepts visuels, tableau 1.4 et chimiques, tableau 1.5. Ces percepts serviront à identifier les défauts visuels (fissure, corrosion et défaut de soudure) en 6 critères (de 20 à 25) ainsi que les défauts chimiques (émanation de gaz) en 5 critères. Les critères des percepts visuels sont les suivants : (20) la réflectance, (21) l'émissivité des corps notée E comprise entre 0 et 1 (en fonction du matériau de l'environnement), (22) la luminosité (23) les motifs répétitifs et (24) le risque de pollution visuelle. Nous avons choisi le matériau principal des différents cas d'application (cas 1 : E Acier polie = 0.04, cas 2 : E Ciment = 0.5 / Calcaire = 0.95, cas 3 : E Brique rouge = 0.9 / Papier ou Panneau d'amiante = 0.94) [111]. Enfin, le dernier critère (25) évalue la difficulté d'acquisition du percept visuel. Cette évaluation permet de définir si un besoin de traitement est nécessaire, les sources de bruits et les stratégies de traitement d'images de ces POI dans les différents cas d'études.

Tableau des critères d'évaluation des percepts visuels								
Catégorie	N°	Critères	Niveau 1		Niveau 2		Niveau 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Critères de percepts visuels	20	Réflectance	1	Faible. Mat.	3	Moyenne. Faible brillance.	9	Fort. Milieu à surface réfléchissante.
	21	Emissivité des corps	1	Faible. Acier poli $E=0,04$	3	Moyenne. Ciment $E=0,5$	9	Fort. Brique rouge $E=0,9$
	22	Luminosité	1	Faible. Entre 1 et 100 lux.	3	Moyenne. Entre 100 et 300 lux	9	Fort. Entre 300 et 2000 lux.
	23	Motifs répétitifs	1	Faible. De 0 à 2 motifs au m ² .	3	Moyenne. 5 motifs au m ² .	9	Fort. Plus de 10 motifs au m ² .
	24	Risque de pollution visuel	1	Faible. Conditions quasi-optimales.	3	Moyenne. Bruit léger dans l'image. Filtre appliqué. Léger post-traitement.	9	Avéré. Particule en suspension. Fumée. Succession de filtres et post-traitement. Appliqué au cas par cas.
	25	Cotation du percept visuel	$\Sigma < 12$	Percept à acquisition facile. Peu de bruit d'image. Pas de post-traitement nécessaire.	$12 < \Sigma < 24$	Percept à acquisition modérée. Source de bruit sur l'image. Post-traitement léger.	$24 < \Sigma < 36$	Perception à acquisition difficile. Source de bruit gênant l'identification d'un point d'intérêt. Post-traitement au cas par cas.

TABLE 1.4 – Tableau d'évaluation des critères contextuels des percepts visuels.

Le cinquième et dernier critère concerne les percepts chimiques 1.5. Les 5 critères (de 26 à 30) sont les suivants : (26) température, (27) pression, (28) Vitesse de l'air, (29) Volume (en fonction des critères de l'environnement Surface (4) et Hauteur (5)). Enfin, le dernier critère (30) évalue la difficulté d'acquisition du percept chimique. Cette approche permet de connaître les sources de bruit et filtres pouvant être utilisés. De plus, cette évaluation est basée sur les valeurs physiques et chimiques de l'environnement, permet

d'orienter sur les choix de capteurs. Dans notre étude, nous focalisons l'attention sur les données d'entrées sensorielles amenant le robot à atteindre la source du gaz.

Tableau des critères d'évaluation des percepts chimiques								
Catégorie	N°	Critères	Niveau 1		Niveau 2		Niveau 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Critères de percepts chimiques	26	Température	1	Température ambiante (20°C).	3	Différence de 10 °C inférieur ou supérieur à la température ambiante.	9	Différence de plus de 20 °C supérieur ou inférieur à la température ambiante.
	27	Pression	1	Pression ambiante (1013,25 hPa)	3	Pression ambiante avec plus de 20 hPa.	9	Pression ambiante avec plus de 50 hPa.
	28	Vitesse d'air	1	Faible 0,2 m/s.	3	Moyenne. 2 m/s.	9	Fort. 20 m/s.
	29	Volume Surface (n°4) + Hauteur (n°5)	$\Sigma < 6$	Volume faible	$6 < \Sigma < 12$	Volume moyen	$12 < \Sigma < 18$	Volume important
	30	Cotation du percept chimique	$\Sigma < 15$	Percept à acquisition facile. Peu de bruit de signal Pas de post-traitement nécessaire.	$15 < \Sigma < 30$	Percept à acquisition modérée. Source de bruit sur le signal. Post-traitement léger.	$30 < \Sigma < 45$	Perception à acquisition difficile. Source de bruit gênant l'identification d'un point d'intérêt. Post-traitement au cas par cas.

TABLE 1.5 – Tableau d'évaluation des critères contextuels des percepts chimiques.

Synthèse

Les tableaux de cotation, présentés précédemment, couvrent ici 5 thématiques majeures qui sont respectivement : l'environnement, l'inspection, le risque, les percepts visuels et les percepts chimiques. Dans chaque thématique, plusieurs critères sont regroupés et évalués. Ces critères sont séparés en 3 niveaux de cotations impliquant un niveau croissant de difficulté : niveau 1, 2 et 3 pour une cotation respectivement associée de 1, 3 et 9. Cette évaluation des critères nous donne des informations sur les choix de recherche et développement à mener, afin d'adapter la suite robotique aux différents cas. Ces évaluations peuvent s'inscrire dans un process plus général pour l'entreprise, ayant pour objectif d'adapter un robot pour plusieurs cas d'inspection plus ou moins complexes et de limiter et concentrer le coût de R&D à un seul prototype, plutôt qu'à un prototype répondant à un seul cas. Dans les prochaines parties, nous présentons l'évaluation de ces différents cas en fonction des critères choisis.

1.2.4 Évaluation des cas d'études

Comme nous l'avons vu dans la section précédente, l'évaluation des critères nous apporte des informations de conception et d'aide à la décision sur les besoins robotiques, les capteurs et les algorithmes. Les points communs et différences entre les différents cas d'études peuvent ainsi être mis en évidence. Elles apportent aussi un cadrage nécessaire pour une meilleure réponse au besoin du sujet de thèse : la reconnaissance automatique de points d'intérêts par un robot d'inspection dans un environnement contraint et dégradé. L'évaluation des cas a été réalisée par l'auteur sous le conseil technique de l'entreprise, spécialisée dans l'inspection industrielle. Dans cette partie, nous présentons les résultats des différentes évaluations des différents cas.

Évaluation des cas liés aux critères de l'environnement

L'évaluation de l'environnement, tableau 1.6 met en relation les critères propres aux lieux, sa géométrie et sa surface par exemple. Les données de sorties de cette analyse nous apportent des informations sur l'amplitude à analyser et d'adapter les capteurs, l'architecture du robot mobile, l'algorithme de navigation et de franchissement d'obstacle en fonction de la nature du lieu.

Catégorie	N°	Critères	Cas 1		Cas 2		Cas 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Environnement	1	Type d'environnement	9	Environnement intérieur exclusivement	9	Environnement intérieur exclusivement	9	Environnement intérieur exclusivement
	2	Matériaux d'environnement / Capteurs du robot	9	Risques de perturbations capteurs avérés.	9	Risques de perturbations capteurs avérés.	1	Risques de perturbations faibles.
	3	Géométrie des lieux	1	Géométrie simple (forme élémentaire).	3	Géométrie élaborée (assemblage d'éléments simples).	9	Géométrie complexe ou déstructurée.
	4	Surface	1	Inférieur ou égale à 250 m².	9	Supérieure à 500 m²	1	Inférieur ou égale à 250 m².
	5	Hauteur	3	Entre 3 et 5 mètres.	9	Supérieure à 5 mètres	1	Inférieur ou égale 3 mètres.
	6	Obstacles	3	Moins de 5 obstacles de moins de 10 cm.	9	Supérieure à 5 obstacles de 10 cm et plus.	9	Supérieure à 5 obstacles de 10 cm et plus.
	7	Dynamique de l'environnement	3	Faibles activités humaines et/ou environnement partiellement stable.	3	Faibles activités humaines et/ou environnement partiellement stable.	9	Activité humaine forte et/ou environnement instable avéré.
	8	Trajectoire nécessaire	1	Navigation en ligne droite possible.	9	Une majorité de points avec un besoin d'anticiper l'itinéraire. Trajectoire contrainte.	9	Une majorité de points avec un besoin d'anticiper l'itinéraire. Trajectoire contrainte.
	9	Cotation d'environnement	30	Environnement à faibles difficultés	60	Environnement à appréhender et robot à adapter.	48	Environnement complexe à fortes contraintes. Test en simulation.

TABLE 1.6 – Tableau d'évaluation des cas sur les critères d'environnement

L'évaluation des critères d'environnement (tableau 1.6) nous montre que le cas 2 est le plus complexe à cause de son volume à inspecter et l'est aussi pour sa durée avec une sollicitation importante. Le cas 1 et 3 partagent des caractéristiques d'espace. Le cas 3 se

démarque par la particularité d'être un cas à faible volume et déstructuré, tableau 1.6. Les capteurs choisis devront être différents du premier et deuxième cas en raison d'une différente topologie, un environnement déstructuré avec des POI potentiellement bloqués ou cachés par les décombres. Notre hypothèse de travail sur les deux premiers cas (ouvrages industriels de stockage d'hydrocarbure et les carrières souterraines) suppose que le robot devra être capable de pouvoir franchir des obstacles, cartographier, naviguer et adapter sa trajectoire en fonction de la configuration du lieu (carte). Les capteurs sensibles aux effets de cage Faraday (premier cas) tels qu'une IMU, par exemple, devront être filtrés pour limiter les risques d'erreurs dans le cas d'une source odométrique. Un capteur visuel à champ large conviendrait pour capturer suffisamment les volumes importants, sous réserve que la locomotion du robot ne génère pas un flux de particules visibles lors de l'acquisition.

Évaluation des cas lié aux critères de l'inspection

L'évaluation du critère d'inspection, tableau 1.7 met en relation les critères d'inspection : les types de points d'intérêts, le niveau technologie, la durée et la récurrence de la mission. Les données de sorties de cette analyse nous apportent des informations sur le choix des capteurs à utiliser pour les points d'intérêts, sur l'autonomie nécessaire du robot, la durée, ainsi que la ré-exploitation des données d'une inspection précédente pour réaliser un comparatif d'évolution dans le cas d'une récurrence. Les résultats sont présentés 1.7.

Tableau des résultats d'évaluation de l'inspection.								
Catégorie	N°	Critères	Cas 1		Cas 2		Cas 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Type d'inspections	10	Types de points d'intérêts	3	2 points d'intérêt. Visuel et chimique par exemple.	3	2 points d'intérêt. Visuel et chimique par exemple.	3	2 points d'intérêt. Visuel et chimique par exemple.
	11	Niveau technologique	1	Nécessite des capteurs peu couteux (inférieur à 50 € unité) disponible sur le marché.	3	Nécessite des capteurs de bonne précision (entre 50 et 200 € unité) avec des librairies en source libre.	9	Nécessite des capteurs spécifiques ou de l'industrie (supérieur à 200 €) avec un kit de développement logiciel constructeur.
	12	Durée	1	Inspection rapide. Moins de 30 minutes	9	Inspection longue. Supérieure à 1 heure.	9	Inspection longue. Supérieure à 1 heure.
	13	Récurrence	3	Nécessite des visites épisodiques (4 visites sur une année) pour un environnement ou une installation à surveillance modéré.	3	Nécessite des visites épisodiques (4 visites sur une année) pour un environnement ou une installation à surveillance modéré.	9	Nécessite des visites récurrentes (supérieures à 4 visites par mois) pour un environnement ou installation avec une surveillance renforcée.
	14	Cotation d'inspection	8	Niveau d'inspection faible. Accessible à la majorité du personnel. Sollicitation faible.	18	Niveau d'inspection moyen. Nécessite du personnel formé. Sollicitation moyenne.	30	Niveau d'inspection fort. Nécessite du personnel qualifié. Forte sollicitation.

TABLE 1.7 – Tableau d'évaluation des cas sur les critères d'inspection.

L'évaluation des critères d'inspection (tableau 1.7) nous montre que le cas 1 (environnement industriel) est le moins contraignant sur le plan technologique et sur la durée. Le

cas 2 (carrière souterraine) nécessite une autonomie importante pour répondre au besoin. Les capteurs et algorithmes nécessaires devront prendre en compte la complexité de l'environnement, notamment pour la recherche de fissures. La récurrence reste la même pour les deux premiers cas, une méthode de comparaison entre deux cartographies espacées dans le temps, peut nous permettre sur le deuxième cas, d'indiquer une présence d'éboulis. Enfin, le cas 3 requiert un niveau technologique supérieur pour détecter des humains sous les décombres, une approche thermique peut être envisagée pour les localiser et une approche visuelle classique permettra de détecter les fissures afin d'apporter des éléments sur la santé structurelle du milieu. La durée et la récurrence sont des éléments très contextuels et dépendent de la situation. Dans la majorité des cas de sauvetage, il est préférable que l'intervention soit rapide, notamment lorsque les pronostics vitaux sont engagés. Cependant, des cas contraints peuvent réduire les possibilités d'accès et ainsi pénaliser le temps d'intervention. La récurrence, dans ce cas, n'est pas annuelle et plutôt en fonction quotidienne ou au besoin, en fonction de l'évolution de la situation. L'objectif est donc d'intervenir et inspecter rapidement avec le plus d'informations fiables et pertinentes, ainsi qu'une récurrence potentiellement importante dans un délai court. Dans ce dernier cas, le robot devra être capable d'évoluer rapidement, d'enlever tout obstacle empêchant sa locomotion et son évolution dans l'environnement et pouvant intervenir une nouvelle fois si les informations ou si la situation le demande.

Évaluation des cas liés au risque

L'évaluation du critère du risque, tableau 1.8 met en relation les critères de risques liés à la mission : la fréquence d'exposition, la gravité et la maîtrise du risque. Les données de sorties de cette analyse nous apportent des informations sur les types de matériaux, la stratégie d'exploration, la redondance des organes électroniques et la stratégie de récupération du robot. Les résultats sont présentés 1.8.

L'évaluation de ces critères (tableau 1.8) nous démontre que le cas 1 est le moins dangereux. Le cas 2 et le cas 3 se démarquent aussi par leur dangerosité, tableau (1.8). Cependant, cette analyse reste à moduler suivant les situations des cas 2 et cas 3. Un cas de destruction d'infrastructure urbaine (cas 3) peut être plus dangereux et complexe à traiter qu'un cas d'inspection de carrière ou mine souterraine. Notre hypothèse de travail se base sur le fait que le robot devra avoir une redondance sur sa locomotion pour les cas 2 et cas 3 avec des matériaux résistants. La stratégie d'exploration devra se faire de manière minutieuse pour le cas 3, afin de réduire l'impact du robot sur l'environnement

Catégorie	N°	Critères	Cas 1		Cas 2		Cas 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Évaluation du risque	16	Fréquence d'exposition Durée (n°12) + Récurrence (n°13)	4	Faible exposition.	12	Exposition moyenne.	18	Exposition fort.
	17	Gravité	3	Faible. Dégâts esthétiques ou superficiels.	9	Fort. Dégâts lourds sur le robot empêchant sa progression de manière définitive.	9	Fort. Dégâts lourds sur le robot empêchant sa progression de manière définitive.
	18	Maîtrise du risque	1	Forse. Présence d'éléments de maîtrise prioritaire et secondaire assurant un haut niveau de prévention et de protection.	3	Moyenne. Présence d'éléments de maîtrise secondaire.	9	Absence de mesure ou mesure inadaptée.
	19	Cotation du risque	8	Risque faible.	24	Risque fort.	36	Risques fort

TABLE 1.8 – Tableau d'évaluation des cas sur les critères de risque.

instable et d'éviter une dégradation du milieu. La maîtrise du risque étant complexe à aborder pour le troisième cas, une stratégie de récupération doit être envisagée.

Évaluation du cas contextuel au percepts visuels

L'aspect vision est une problématique connue dans la robotique mobile. L'évaluation du critère du percept visuel, tableau 1.9 met en relation les critères d'acquisition du percept visuel : une approche basée sur la lumière, la potentielle pollution visuelle et les motifs répétitifs impactant la qualité des reconstitutions en 3D. Les données de sortie de cette analyse nous apportent des informations sur la stratégie d'acquisition et traitement d'images. Les résultats sont présentés dans le tableau 1.9. La quatrième catégorie concerne les percepts visuels, tableau 1.9.

Catégorie	N°	Critères	Cas 1		Cas 2		Cas 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Critères de percepts visuels	20	Réflectance	1	Faible. Mat.	3	Moyenne. Faible brillance.	1	Moyenne. Faible brillance.
	21	Émissivité des corps	3	Moyenne. Acier inoxydable 301 E=0,54	9	Fort. Granite E=0,971	9	Fort. Peinture blanche E=0,77
	22	Luminosité	1	Faible. Entre 1 et 100 lux.	1	Faible. Entre 1 et 100 lux.	3	Moyenne. Entre 100 et 300 lux
	23	Motifs répétitifs	3	Moyenne. 5 motifs au m².	9	Fort. Plus de 10 motifs au m².	9	Fort. Plus de 10 motifs au m².
	24	Risque de pollution visuel	3	Moyenne. Bruit léger dans l'image. Filtre appliqué. Léger post-traitement.	3	Moyenne. Bruit léger dans l'image. Filtre appliqué. Léger post-traitement.	9	Fort. Particule en suspension. Fumée. Succession de filtres et post- traitement. Appliqué au cas par cas.
	25	Cotation du percept visuel	11	Percept à acquisition facile. Peu de bruit d'image. Pas de post- traitement nécessaire.	25	Perception à acquisition modérée. Source de bruit sur l'image. Post-traitement léger.	31	Perception à acquisition difficile. Source de bruit gênant l'identification d'un point d'intérêt. Post-traitement au cas par cas.

TABLE 1.9 – Tableau d'évaluation des cas sur les critères de percepts visuels.

L'évaluation des critères de percepts visuels (tableau 1.9) nous montre un score su-

périeur pour les carrières pour le cas 3. Le cas 1 reste le cas le plus perceptible sur cet aspect-là aussi. Le cas 3 a une cotation forte en raison du risque de pollution visuelle et de l'émissivité, (tableau 1.9) pouvant impacter la qualité d'acquisition visuelle. Notre hypothèse de travail se base sur le fait que les cas 2 et 3 devront être potentiellement traité à partir de capteurs visuels redondants (stéréoscopie) et/ou avec un capteur laser permettant d'obtenir un nuage de points pour avoir des informations sur la distance des POI visuels et avoir une redondance dû aux contraintes de l'environnement sur cette thématique.

Évaluation du cas contextuel au percepts chimiques

Les robots sont des outils intéressants pour la recherche de source de gaz [164] dans les environnements confinés grâce à leur capacité de franchissement sur les obstacles et l'intelligence embarquée. Ils permettent de s'approcher au plus proche de la zone critique et de donner des informations importantes pour la maintenance ou pour la prise de décision en situation de crise. Cependant, la réalité du terrain peut entraîner des achoppements et peut pénaliser le succès de la mission (bruit de captation, environnement hostile, obstacles imprévus, sol impraticable, etc.).

L'évaluation du critère du percept chimique, tableau 1.10 met en relation les critères d'acquisition du percept chimique : une approche basée sur la loi des gaz parfaits est proposée. L'évaluation porte sur la température, la pression, la vitesse de l'air ambiant et le volume. Cette approche nous permet d'appréhender la situation physico-chimique de l'environnement et d'adapter les capteurs chimiques et les algorithmes de recherche de source de gaz en fonction des cas. Les résultats sont présentés 1.10.

Tableau des résultats d'évaluation des percepts chimiques.								
Catégorie	N°	Critères	Cas 1		Cas 2		Cas 3	
			Cotation	Critère associé	Cotation	Critère associé	Cotation	Critère associé
Critères de percepts chimiques	26	Température.	3	Différence de 10 °C inférieur ou supérieur à la température ambiante.	9	Différence de plus de 20 °C supérieur ou inférieur à la température ambiante.	1	Température ambiante (20 °C).
	27	Pression	3	Pression ambiante avec plus de 20 hPa.	3	Pression ambiante avec plus de 20 hPa.	1	Pression ambiante (1013,25 hPa)
	28	Vitesse d'air.	1	Faible 0,2 m/s.	3	Moyenne. 2 m/s.	1	Faible 0,2 m/s.
	29	Volume Surface (n°4) + hauteur (n°5)	4	Volume faible	18	Volume important.	2	Volume faible.
	30	Cotation du percept chimique	11	Percept à acquisition facile. Peu de bruit de signal Pas de post-traitement nécessaire.	33	Perception à acquisition difficile. Source de bruit gênant l'identification d'un point d'intérêt. Post-traitement au cas par cas.	5	Percept à acquisition facile. Peu de bruit de signal Pas de post-traitement nécessaire.

TABLE 1.10 – Tableau d'évaluation des cas sur les critères de percepts chimiques.

1.2.5 Synthèse

L'évaluation nous montre que le cas 3 et le cas 1 seront des cas plus perceptibles chimiques grâce à leur faible volume (lien avec les critères (4) surface et (5) hauteur du tableau 1.6). Le cas 2 est un volume important et complexe pouvant amener le robot à perdre le panache de gaz [128] dans une configuration avec des couloirs, des obstacles, des

flux d'air ascendants, par exemple.

Notre hypothèse de travail se base sur une approche par la concentration de gaz ainsi que la vitesse et l'orientation du flux d'air. Des traitements de données par filtrage seront à prévoir dans un second temps, pour améliorer les résultats et apporter de meilleures précisions. Le rafraîchissement de nouvelles valeurs des capteurs chimiques devra prendre en compte la vitesse de locomotion pour l'acquisition. Un robot trop rapide risque de générer un flux d'air et donc de la pollution sur l'acquisition. De plus, si le capteur à un taux de rafraîchissement trop bas, il prendra la même valeur sur son trajet, car le capteur ne sera pas à jour sur la nouvelle valeur.

1.2.6 Expression des besoins

Les besoins sont exprimés par la figure 1.8. Le service du produit dépendra du business modèle choisi par l'entreprise. La vente ou la location du produit profitera aux entreprises d'inspection, de prestations ou dans le domaine militaire afin d'achever des services de reconnaissances spécifiques. Autrement, l'entreprise Pilgrim pourra utiliser le produit pour fournir une prestation d'inspection robotisée dans les milieux dégradés et confinés.

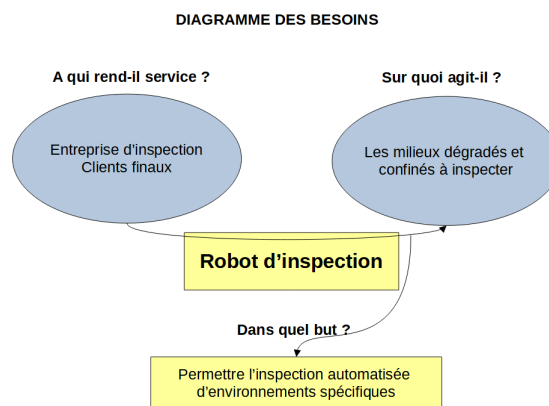


FIGURE 1.8 – Diagramme "Bête à cornes" pour la définition du besoin.

Nous l'avons vu précédemment, l'évaluation des trois cas d'environnement présente des contraintes. Celles-ci sont le résultat des interactions entre le robot d'inspection et les différents critères d'environnement, illustré par la figure 1.9.

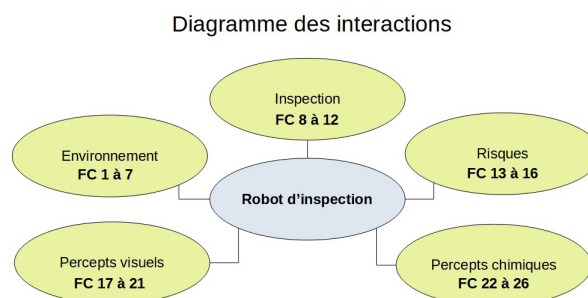


FIGURE 1.9 – Diagramme des interactions avec les critères de l'environnement.

Au vu des résultats de cotation, nous avons listé les différentes contraintes comme données d'entrées pour notre étude :

N°	Thèmes	Contraintes d'environnement
FC 1	Environnement	Environnement intérieur
FC 2	Environnement	Risques de perturbations de capteurs.
FC 3	Environnement	Géométrie complexes
FC 4	Environnement	Grand espace (500 m ² , 5 m de haut)
FC 5	Environnement	Obstacle supérieurs à 10 cm (20 cm)
FC 6	Environnement	Environnement instable.
FC 7	Environnement	Capable d'appréhender des environnements complexes avec des obstacles, des inclinaisons, des terrains non homogène.
FC 8	Inspection	2 points d'intérêts à détecter : visuel et chimique
FC 9	Inspection	Nécessite des capteurs spécifiques
FC 10	Inspection	Inspection d'une durée de 1 heure ou plus
FC 11	Inspection	Nécessite des visites récurrentes
FC 12	Inspection	Niveau d'inspection élevé
FC 13	Risque	Exposition forte
FC 14	Risque	Gravité forte : risque de dégâts importants sur le robot
FC 15	Risque	Maîtrise du risque : absence de mesure ou mesure inadapté
FC 16	Risque	Récurrence avérée
FC 17	Percepts visuels	Réflectance moyenne
FC 18	Percepts visuels	Émissivité forte
FC 19	Percepts visuels	Luminosité faible
FC 20	Percepts visuels	Motifs répétitifs avec une forte occurrence
FC 21	Percepts visuels	Cotation du percept visuel : perception à acquisition difficile
FC 22	Percepts chimiques	Température : différence de température importante
FC 23	Percepts chimiques	Pression : différence de pression moyenne
FC 24	Percepts chimiques	Vitesse du flux d'air : Moyenne
FC 25	Percepts chimiques	Volume important
FC 26	Percepts chimiques	Perception à acquisition difficile

1.3 Conclusion de fin de chapitre

La méthode d'analyse des cas nous a permis de faire le lien entre l'environnement et le robot pour définir les contraintes et les différentes interactions. L'évaluation des cas d'étude, des tableaux 1.6 (l'environnement), 1.7 (l'inspection), 1.8 (le risque), 1.9 (les percepts visuels), 1.10 (les percepts chimiques), nous montre que les environnements, inspections et risques les plus critiques sont le cas 2 (carrières et mines souterraines) et le cas 3 (scénario de post-accident). La surface des lieux et la hauteur des lieux sont des critères impactant la cotation du cas 3. Nous avons présenté ci-dessus une approche de cotation de cas d'inspection dont l'objectif est de définir les caractéristiques les plus notables et à appréhender dans l'approche d'inspection à adopter, les contraintes de locomotions à prévoir et les capteurs nécessaires pour optimiser la perception de l'environnement tout en évitant, limitant ou en supprimant les éventuels bruits de captation. Cette étude oriente l'état de l'art sur la recherche de la plateforme robotique, des capteurs et des algorithmes de détection des POI visuels et chimiques avec la liste des contraintes présentées précédemment. L'état de l'art et les différents choix sont décrits dans le Chapitre II.

ÉTAT DE L'ART

Dans le chapitre 2, une analyse des éléments contextuels a été réalisée des différents cas d'inspection. Ces cas d'inspection concernent les environnements industriels et les équipements navals pour le cas 1, les carrières souterraines pour le cas 2 et les situations de post-accidents pour le cas 3. Cette analyse répond à la problématique de réaliser un prototype robotique pouvant répondre à ces 3 cas d'application. Il est souligné l'importance d'avoir une méthode d'inspection automatisée, adaptée aux POI, anticipant les nuisances visuelles et chimiques de l'environnement. Ces informations précieuses permettront de définir une plateforme robotique robuste, capable de franchir des obstacles et d'évoluer dans un environnement avec un sol hétérogène, non-linéaire et déformé.

Dans ce chapitre, nous aborderons en premier lieu les inspections industrielles en tant que domaine technique. En seconde partie, un état de l'art et une définition de l'architecture robotique est proposée à travers une matrice de cotation, en prenant en compte les données de sorties de l'analyse des éléments contextuels du chapitre 2. Dans la troisième partie, nous faisons un état de l'art sur l'architecture choisie, spécifique à la catégorie du robot mobile. Les quatrième et cinquième parties se concentrent davantage, respectivement, sur la détection des points d'intérêts visuels et chimiques. Un état de l'art des algorithmes de détection et d'asservissement est évoqué dans ces deux dernières parties.

2.1 État de l'art : Inspection industrielle

Cette section va essentiellement relater des travaux en lien sur la robotique mobile afin de définir une plateforme adaptée aux environnements définis précédemment. Nous évoquerons le cas des robots marchants, offrant des avantages sur les robots classiques (à roue, à chenille, etc.). Par la suite, nous évoquerons les méthodes de détection des points d'intérêts visuels et chimiques. Enfin, cette dernière section exposera les différents travaux concernant l'asservissement visuel et chimique.

2.1.1 Introduction

Depuis environ 30 ans [129], les robots et l'inspection ont été étroitement liés pour faciliter les opérations de contrôles techniques d'ouvrages, d'infrastructures ou d'environnements complexes. Nous décrivons ici les méthodes d'inspections usuelles. Habituellement réalisées par des humains, les inspections industrielles mobilisent le savoir-être, savoir-faire et connaissances techniques liées aux produits à inspecter et à l'environnement industriel dans lequel ils évoluent. Depuis plusieurs années, les robots tendent à prendre le relais sur ce type d'opération. Les inspections industrielles ont pour but, en général, de réaliser l'interface entre le client final et le producteur industriel dans l'optique de vérifier, qualifier la conformité réglementaire et viabiliser les installations. Les non-conformités ou anomalies identifiées sont renseignées sur les supports et rapports de contrôle. Le client est donc informé et l'inspecteur peut ainsi conseiller et prescrire des solutions techniques pour appliquer des actions correctives, préventives ou des améliorations des équipements.

2.1.2 Les méthodes d'inspections visuelles

Les inspections visuelles sont régies par la norme EN ISO 17637 [6]. Dans le domaine de la métallurgie, elles sont réalisées dans le cadre de l'épreuve de qualification des soudeurs selon la norme EN ISO 9606-1 [7] dans le cadre de contrôles non destructifs des assemblages soudés. Les inspections visuelles sont réalisées avec un niveau d'éclairage suffisant (supérieur à 350 lux.). Un éclairage conforme permet d'observer et mesurer les défauts de porosités ou de fissures.). Il existe d'autres outils permettant de faire des inspections visuelles comme le calibre de soudage (VT) permettant de mesurer différentes côtes des cordons de soudure. On peut noter que d'autres approches plus modernes et ayant fait ces preuves à base d'IA (supervisée [8] et non supervisée [23]), de vision par ordinateur

(filtre "canny edge" [24] pour la détection de contours) ou d'autres approches d'analyse de structures d'images (morphologie mathématique) [25].

Les méthodes d'inspection de surface

L'inspection de surface [130] permet de vérifier l'état structurel d'une infrastructure. Les surfaces métalliques sont particulièrement impactées par les défauts surfaciques. Nous allons voir ici les méthodes d'inspection pour qualifier la surface, tableau 2.1.2.

Type de méthode	Description
Caméra / Visuel / Thermique [21]	Le plus couramment utilisé dans la robotique mobile. Permet la recherche de formes anormales dans la surface visuelle. Les aides comprennent les baroscopes, les endoscopes ou les caméras (monoculaires, stéréoscopiques, thermiques). Ces capteurs ne détectent que les défauts de surface et dépendent de l'éclairage. Une surface propre est nécessaire et convient aux balayages de zone.
Courant de Foucault [12]	Les défauts tels que les fissures modifient la conductivité électrique, qui modifie à son tour l'impédance de la bobine de mesure. Cette technologie [12] est peu coûteuse et applicable à travers une peinture normale. Elle est sensible aux petits défauts, mais prend du temps pour les grandes surfaces. Cette méthode n'est applicable qu'aux matériaux conducteurs et aux mesures ponctuelles.
Ultrasons [8]	Les ondes sonores ultrasoniques sont réfléchies au niveau des défauts du matériau (détection des anomalies sonores) et des interfaces (mesure de l'épaisseur des parois) et le modèle de réponse est évalué pour déterminer le temps de vol. Il faut un milieu de couplage. Le couplage peut devenir superflu lorsque les méthodes laser ou ultrasoniques deviennent déployables. C'est une méthode de type ponctuel.
Thermographie [10]	La dilatation du gaz refroidit le voisinage d'une fuite. Il est possible de détecter les fuites de liquide provenant de tuyaux dont la température est différente de la température ambiante [10]. Seuls sont détectables les défauts qui affectent la température de surface. Convient aux balayages de surface.

Type de méthode	Description
Fuite de courant magnétique [9]	Le flux magnétique est pressé hors d'une paroi métallique par toute diminution de l'épaisseur de la paroi. Elle convient pour le balayage de zones. La fuite du flux étant un vecteur, on ne peut mesurer qu'une seule direction.
Inspection par particule magnétique [11]	Si un objet magnétisé se fissure, il en résulte des aimants séparés de part et d'autre de la fissure, ce qui entraîne l'accumulation de particules de fer au niveau de celle-ci. Cette méthode est applicable aux matériaux ferromagnétiques.
Radiographie [13]	L'absorption des rayonnements (rayons X ou gamma) dépend de l'épaisseur et de la densité du matériau pénétré. Elle nécessite des réglementations de sécurité coûteuses, mais des systèmes de portiques de lumière sont disponibles depuis peu.
Émission acoustique [14]	Détecte les ondes acoustiques qui sont émises pendant la croissance des défauts microscopiques. Cette technique passive est sensible uniquement aux processus actifs lors du contrôle. L'équipement doit être sous pression. Cette méthode est utilisée pour les essais hydrauliques ou gaz comprimés.
Pénétrant liquide [15]	Un liquide (souvent chargé d'un colorant fluorescent) est appliqué sur l'objet et s'infiltré dans les défauts de surface. Le liquide qui s'est infiltré est ensuite dessiné et rendu visible. Cette méthode est facile à appliquer sur de grandes surfaces. Elle permet de manipuler facilement des géométries complexes.

Les méthodes de détection de fuite

La précision des algorithmes de recherche de source de gaz dépendent particulièrement des capteurs choisis et de leurs caractéristiques (précision, temps de réponse, types de gaz détectés, coût, etc.). Nous évoquons les différentes techniques pour détecter les fuites de gaz, tableau 2.1.2.

Type de méthode	Description	Réf.
Électrochimique	De nombreux principes différents dont le thermo-catalytique, le thermo-calorimétrique, le semi-conducteur à oxyde métallique et la microbalance à cristal de quartz permettent la détection de fuite chimique. Les capteurs sont souvent conçus pour être sensibles à un seul gaz ou à un mélange combustible.	[16]
Acoustique	Une fuite de gaz provoque un bruit au point de rejet en raison des tourbillons émergents. Une petite fuite de liquide provoque un bruit dû à la cavitation, qui peut être détecté par un hydrophone ou un transducteur à l'extérieur de la conduite.	[17]
Thermographie	La dilatation du gaz refroidit le voisinage d'une fuite. Il est possible de détecter les fuites de liquide provenant de tuyaux dont la température est différente de la température ambiante. Elle permet une inspection à distance.	[18]
Electro-optique	Les gaz absorbent le rayonnement électromagnétique de manière caractéristique et sont donc observables par leur empreinte spectrale. Il s'agit généralement de systèmes basés sur un laser IR et permettant une inspection à distance.	[19]
Basé sur un modèle	Les valeurs mesurées sont comparées aux prédictions du modèle thermodynamique. Elle est utilisée uniquement pour les tuyauteries en raison de l'effort de modélisation.	[20]

2.1.3 Synthèse générale : choix des capteurs

Concernant les choix de capteurs, nous prévoyons l'utilisation de caméra pour les points d'intérêts visuels et les capteurs de gaz oxyde métallique pour les anomalies chimiques. Les contraintes du robot, en matière de poids embarqué et de puissance de calcul, seront à prendre en compte. L'avantage de la caméra est de pouvoir balayer un maximum de zones et d'identifier les anomalies visuelles. Couplés à un capteur de profondeur, les phénomènes visibles peuvent alors être placés dans le repère du robot (la distance entre le point d'intérêt et le robot) et replacés sur le repère de la carte (par rapport à l'em-

placement du robot dans le repère carte). L'inconvénient est que les capteurs visuels sont sensibles aux conditions lumineuses. La particularité des milieux contraints et dégradés est dans la majorité des cas une absence de qualité lumineuse pour des opérations d'analyses visuelles minutieuses. Pour régler le problème, un dispositif lumineux composé de LED normalisée pour l'inspection visuelle est prévu et l'algorithme devra prendre en compte les contraintes de l'environnement pour la génération d'un modèle. L'utilisation de capteurs en oxyde de métal est un choix discutable : les capteurs sont abordables (15€ environ l'unité), détectent plusieurs gaz par la concentration et s'intègrent facilement dans les architectures robotiques. L'utilisation d'une caméra spécifique comme les caméras multispectrales ou chimiques simplifieraient le processus en utilisant une méthode de jacobien visuel applicable sur les 2 percepts. Au moment où le projet a débuté, la question s'est posée de l'utilisation de tels dispositifs. Il s'est révélé que le prix (avoisinant 10-20 K€) et le poids (1,4 kg) (caméra chimique [131]) était inabordable et irréaliste pour l'envergure du robot souhaité par l'entreprise et sur la cohérence du traitement des cas d'application (un robot de grande envergure n'est pas souhaitable pour les milieux contraints.).

2.2 État de l'art : La plateforme robotique terrestre

Dans cette section, nous allons analyser différentes architectures robotiques proposées dans la littérature. Ces plateformes partagent plusieurs domaines d'utilisation : l'exploration de zones nucléaires, spatiales, sous-terrain-es, industrielles, partagent les mêmes caractéristiques environnementales (Chapitre 2, Section 2.1 : Les environnements complexes), ces types d'environnements sont difficiles d'accès, dangereux pour les humains et impliquent une stratégie d'exploration, donc de mobilité et de locomotion.

2.2.1 Introduction

La robotique mobile est par définition des robots dotés de locomotion et souvent appelé AMR (Autonomous Mobile Robots). Du chien électrique conçu par John Hammond et Benjamin Miessner (1915), en passant par les tortues cybernétiques de William Grey Walter (1950), ou encore le renard électronique d'Albert Ducrocq (1953), le monde de la robotique mobile a toujours été intimement lié au monde du vivant et à la compréhension que l'on a de celui-ci [132]. Dans cette partie, nous allons évaluer les différentes possibilités d'architectures de la robotique mobile.

Approche de l'étude : La classification et l'évaluation

Aujourd'hui, le monde de la robotique mobile s'est diversifié et divisé en de multiples catégories comprenant de multiples formes et stratégies de mobilité. La diversité de ces approches est notable par l'utilisation de principes de systèmes mécaniques mobiles d'horizons diverse (approche biomimétique, simplistes, hybrides, aérodynamique, géométrique, etc.). Ces différentes approches peuvent être réparties en trois grandes familles : les systèmes aériens, les systèmes terrestres et aquatiques [115]. Dans cette étude, nous allons considérer les systèmes aquatiques hors de propos dû à la typologie de nos cas. Les drones volants ont une mobilité intéressante ainsi que la possibilité offerte d'obtenir différents points de vue par la (les) caméra (s) embarquée (s) sont intéressantes, néanmoins le soulèvement des particules par la rotation des hélices ainsi que le risque de dispersion des gaz à détecter pour les mêmes raisons sont rédhibitoires dans ces environnements de type intérieur.

Attendu dans notre cas la nécessité d'une capacité d'emport de charge importante, le

risque de rupture de liaison radio en fonction des environnements, ainsi que les différents problèmes de soulèvement de poussière liés aux drones volants, la recherche a ici été axée plus spécifiquement sur les drones terrestres. Cet état de l'art traitera donc uniquement les architectures évoluant sur la terre ferme. Les robots mobiles peuvent se répartir en 6 catégories : les robots à roues, à traction, marchants, hybrides, coulissants, à roulements. Les différentes catégories sont représentées par des différents prototypes issus de la littérature. Suite à ces différentes présentations d'architectures, un bilan global est réalisé. Enfin, un tableau de notation pour chacune des catégories a été mis au point afin de déterminer le robot terrestre le plus à même de répondre aux besoins de cette étude.

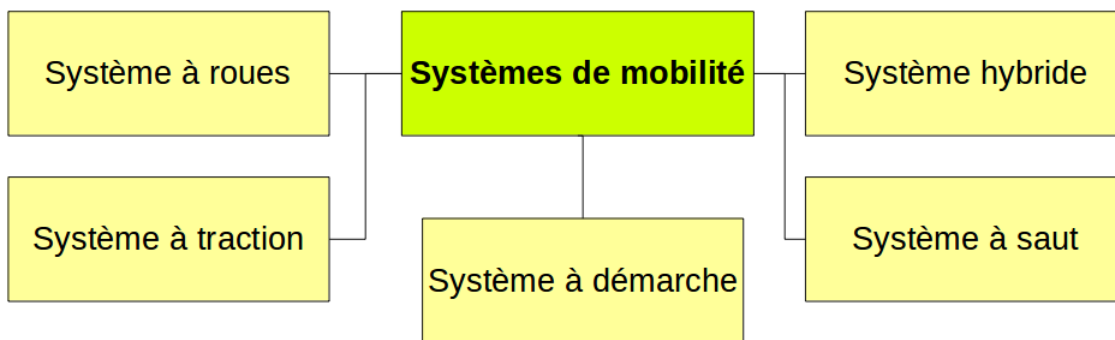


FIGURE 2.1 – Tableau de la robotique mobile terrestre, inspiré des travaux de [115]

2.2.2 La notation

Les critères de notation sont évalués sur 5 niveaux croissants (1 : minimum, 5 : maximum) en fonction des critères choisis. Les éléments sont issus des travaux de [115] et sont composés de 11 critères décrivant les capacités et le mode de locomotion, l'appréhension des sols dégradés, les capacités de franchissement, le niveau de complexité mécanique, la résilience en cas de dysfonctionnement (redondance), efficacité énergétique, la capacité d'emport et la maturité de la technologie disponible. La dissipation de l'élément mobile correspond à l'enfoncement du sol par le moyen de locomotion. La notation opérée par le tableau est consultable ci-dessus (figure 2.1). Cette notation sera utilisée à travers les différentes catégories de robots mobiles terrestres.

Description	1	2	3	4	5
Capacité de translation et rotation au même moment.	Capacité de translation simple.	Capacité de translation simple sur un sol dégradé.	Capacité de rotation et translation de manière asynchrone.	Capacité de rotation et de translation de manière synchrone.	Capacité de rotation et de translation de manière synchrone sur un sol dégradé.
Tolérance à la faute	Très faible.	Faible.	Moyenne.	Forte.	Très forte.
Capacité de se déplacer sur des terrains plats à une vitesse de 40 cm/s.	Jusqu'à 10 cm/s.	Entre 10 et 20 cm/s.	Entre 20 et 30 cm/s.	Entre 30 et 40 cm/s.	Supérieur à 40 cm/s.
Capacité d'escalade progressive qui équivaut à la taille de l'élément mobile. (Diamètre de la roue, longueur des tibia et fémur.)	Jusqu'à 10 cm.	Entre 10 et 15 cm.	Entre 15 et 20 cm.	Entre 20 et 30 cm.	Supérieur à 30 cm.
Capacité de montée en pente d'une inclinaison de 30°.	Jusqu'à 10°.	Entre 10 et 20°.	Entre 20 et 30°.	Entre 30 et 45°.	Supérieur à 45°.
Dissipation de l'élément mobile pendant le mouvement sur le sol.	Très faible.	Forte.	Moyenne.	Faible.	Très faible.
Niveau de complexité mécanique.	Très forte.	Forte.	Moyenne.	Faible.	Très faible.
Capacité à exécuter et à poursuivre le mouvement après dysfonctionnement des organes de mobilités.	Pas de redondance.	Faible	Moyenne.	Fort.	Très fort.
Capacité de déplacement avec une consommation de batterie moindre par distance unitaire.	Très forte.	Forte.	Moyenne.	Faible.	Très fort.
Capacité d'emport compris en 8 à 20 % du poids à vide.	Jusqu'à 8 % du poids à vide.	Entre 8 et 11 % du poids à vide.	Entre 11 % et 14 % du poids à vide.	Entre 14 % et 17 % du poids à vide.	Supérieure à 17 %.
Influence du contact avec le sol pendant la locomotion.	Très forte.	Forte.	Moyenne.	Faible.	Très faible.
Maturité de la technologie disponible.	TRL 1 ou 2.	TRL 3 ou 4.	TRL 5.	TRL 6 ou 7.	TRL 8 ou 9.

TABLE 2.1 – Tableau de notation des différents types de plateformes de la robotique mobile

La première catégorie : Les systèmes à roue.

Les robots à roues constituent la principale et plus répandue catégorie de robots mobiles. On retrouve ce type d'architecture dans diverses applications : industriel, service, santé, transport, etc. La première catégorie regroupe donc les systèmes à roues utilisant deux (Axel rover [116]), quatre (Venus rover [36]), six (Mars Exploration Rover : Opportunity [35]), un rover holonome roues rigides ou roues dites "tentacules" [118].

Les robots à roues ont été très utilisés dans l'exploration de milieu complexe. Les travaux de Seeni [115] sur l'état de l'art de la robotique mobile pour l'exploration en surface de planète (des milieux risqués) montre que 70 % des robots envoyés entre 1970 et 2003 sont des architectures de robots à roues. Les systèmes à roue, figure 2.2, ont l'avantage d'être déjà très présents sur le marché de la technologie, d'être simple, peu coûteux. Cette architecture permet une vitesse importante avec un ratio de consommation d'énergie intéressant. Néanmoins, leur configuration les rend limités en termes de manœuvrabilité. Les systèmes à roues sont dépendants de la qualité du sol à cause des liaisons continues de la locomotion.

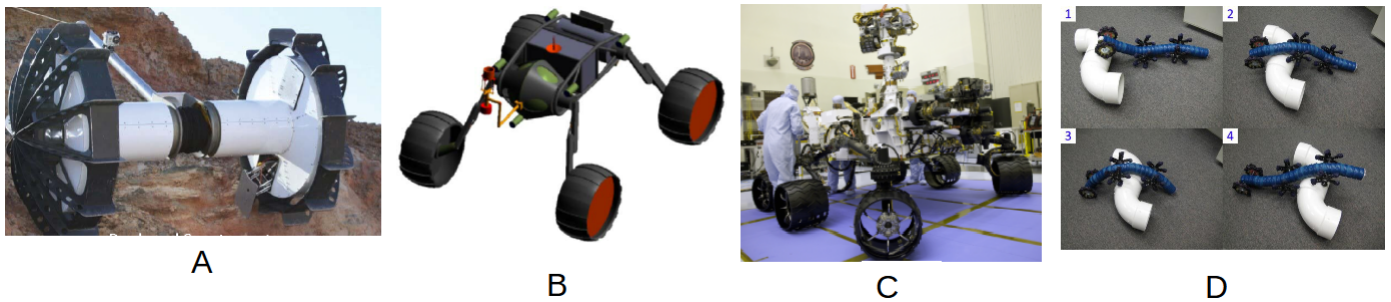


FIGURE 2.2 – Les robots à roues pour les milieux risqués et complexes. A : Axel Rover [116], B : Venus Rover [117] [36], C : Opportunity Rover [35], D : Rover holonome avec des roues "tentacules" [118]

Le concept radical de l'Axel Rover [116] se décrit comme deux véhicules robotiques 2.2 (A) fonctionnant comme un seul : des rovers modulaires, reconfigurables et autoréparables. Ces systèmes modulaires pouvaient s'assembler et former ainsi une chaîne. Le défi de conception le plus important pour le rover Venus 2.2 (B) [117] [36] est la température élevée de l'atmosphère de Vénus (462°C) et la pression (92 bars) en surface. Le robot est conçu avec 4 roues, 8 moteurs pour l'entraînement et la direction, vitesse maximale de 1 cm/s et 4 amortisseurs sur le châssis. Il est aussi équipé de bras de prélèvement/inspection/creusage, perceuse de 2 m avec une capacité de forage à quatre emplacements. Opportunity 2.2 (C) [35] était le deuxième des deux rovers lancés en 2003 à atterrir sur Mars et à commencer à traverser la planète rouge à la recherche de signes d'eau ancienne. Le rover a exploré le terrain martien pendant près de 15 ans, dépassant de loin sa mission prévue de 90 jours. Ce robot 2.2 (D) [118] a un "cou" flexible qui lui permet de se plier autour des obstacles et une méthode de locomotion composée de roues avec des "tentacules" recouvertes de latex qui aident le robot à saisir et à franchir efficacement les obstacles. Cette flexibilité permet au robot de s'adapter à des terrains très accidentés et très difficiles.

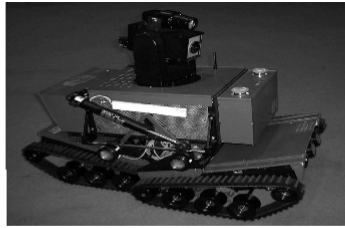
La deuxième catégorie : Les robots à chenille.

La deuxième catégorie concerne les robots à traction, ou plus communément appelé robots à chenille. Les systèmes à traction ont l'avantage d'une bonne capacité de franchissement d'obstacles, bien que limité par la taille des chenilles et la puissance de traction. Néanmoins, leur complexité mécanique est faible et leur forte interaction avec le sol peut s'avérer problématique sur des terrains à pente ascendante ou avec des coefficients de

frottement statique/dynamique faible. Dans cette catégorie, nous retrouvons des configurations à 2 chenilles (Nanokhod [37]), 4 chenilles ("double track mobile robot" [120]) , 8 chenilles par segment (OmniTread Serpentine robot [121]). Ces robots sont présentés en figure 2.3.



A



B



C

FIGURE 2.3 – A : Nanokhod [37], B : Robot à 4 chenilles [120], C : Robot multisegments à 8 chenille : OmniTread

Le rover Nanokhod, Figure 2.3 (A) [37] est conçu pour que les instruments de détection scientifiques puissent être logés dans la cabine de charge utile centrale. Le contrôle et la navigation du rover sont semi-autonomes : il exploite un modèle numérique 3D d'élévation du terrain, acquis au moyen d'une caméra panoramique sur l'atterrisseur. La régulation thermique est entièrement passive tandis que la consommation électrique de pointe est de 3 W. La masse totale de la charge utile est de 1100 g. La masse du rover sans chargement est de 1450 g. Le robot [120], Figure 2.3 (B) [120], se compose d'un mécanisme d'entraînement (les chenilles) et le corps. Le corps se compose d'un bloc de commande et d'une caméra. Le mécanisme est composé de cadres avant et arrière - c'est pourquoi on appelle ce système à double chenille. Le corps est relié à la double chenille par quatre joints. Il est conçu pour que le corps soit incliné en fonction du changement de l'angle relatif entre les cadres avant et arrière. Un amortisseur hydraulique est fixé de chaque côté du corps afin d'absorber les impacts des reliefs accidentés. Le robot OmniTread, Figure 2.3 (C) [121] possède des chenilles utilisé comme "peau" pour le robot. Son architecture en serpentini lui permet donc de se mouvoir comme le font les serpents, c'est-à-dire par mouvement d'ondulation du corps. Il peut ainsi s'affranchir facilement des obstacles complexes avec une charge importante, mais son architecture reste complexe et énergivore.

La troisième catégorie : Les robots sauteurs.

Les robots sauteurs 2.4 sont très peu manœuvrables, et ne peuvent être autonomes qu'avec une certitude et un contrôle précis et répétable des sauts [123], principale locomotion de cette catégorie. Les robots sauteurs peuvent être de très petite taille (5cm) et d'architecture simple, comme nous montre les travaux [122]. Les récents travaux de (Rogers2018), montre des capacités de sauts différentes à travers une extension complète de patte pour un robot sauteur, le "BounceBot".

Dans les travaux de [122], le prototype présenté, Figure 2.4 (A), mesure 5 cm de hauteur pour 7 grammes. Il est capable de faire des sauts pouvant faire 27 fois sa propre taille, soit une hauteur de saut de 1,4 mètre. Il est doté d'un récepteur infrarouge pour la partie commande et d'une batterie. Il met 1 seconde pour fléchir les jambes afin de préparer son saut et 24 ms pour déclencher l'impulsion du saut. Malgré ces performances dynamiques, le robot ne peut naviguer de manière autonome dû à l'absence de capteur. Les travaux de [122] se porte davantage sur l'architecture du robot et vers une architecture hybride, combinant la capacité de saut et de pouvoir planer vers une localisation désirée.

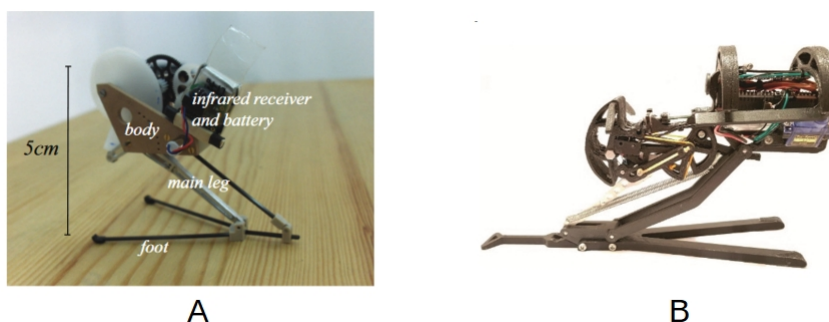


FIGURE 2.4 – A : Micro robot à saut [122], B : BounceBot (Salto) [124]

Les travaux de [124] présente un prototype de robot à saut issu du biomimétisme : le criquet, Figure 2.4. Le robot est composé d'un servomoteur DC pour l'impulsion, d'une caméra, d'un contrôleur moteur, d'un micro-contrôleur (Arduino Uno) et d'une batterie. Il peut sauter 10 fois sa taille accroupie. Si le robot agit de manière cohérente, les surfaces glissantes altèrent les performances de saut en impliquant un léger déplacement vertical (déviations de -20% avec une surface en verre par rapport à un tapis en caoutchouc). L'observation réalisée indique que la vitesse de rotation du robot n'a pas notablement changé.

La quatrième catégorie : Les robots hybrides

Les robots hybrides roulant/marchant (comme ANYmal, [125] et roulant/sauteur (comme Ascento, [126]) présentent un fort intérêt en termes de vitesse et capacités de franchissement, toutefois, ils ont les inconvénients d'une forte interaction avec le sol et d'un niveau d'accès à la technologie complexe (développement de deux systèmes). Le changement de mode de locomotion implique un développement spécifique sur 2 technologies de locomotion, de planification de trajectoire, d'odométrie, etc. Ces robots sont illustrés Figure 2.5.

Les travaux de [126] présentent un robot, Ascento, pouvant opérer dans un monde réel, à l'aide d'un design mécanique basé sur deux roues parallèles avec une capacité de saut élastique, Figure 2.5 (A). Le robot peut aussi s'incliner par mouvement pendulaire pour s'équilibrer lors de virage ou de terrains difficiles. Le prototype est équipé d'un ordinateur embarqué, de 2 batteries (PC et moteurs), d'encodeur de roues, d'une IMU, d'un capteur de distance et d'un système de contrôle par gestes. La capacité du robot à récupérer d'un état de chute est une des principales problématiques de ce dispositif. La procédure de récupération nécessite au moins 2 mètres d'espace et les expérimentations sont faites en

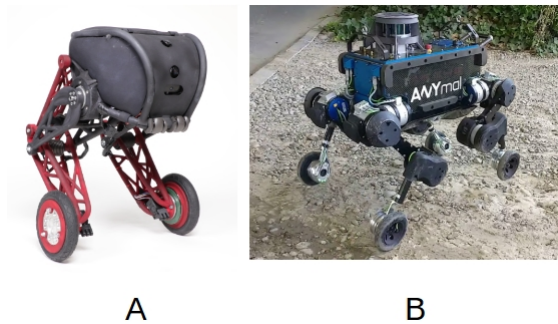


FIGURE 2.5 – A : Ascento, robot hybride roulant et à saut [126], B : ANYmal, robot hybride marchant [125]

extérieurs. Des travaux supplémentaires permettront d'adapter son utilisation à des environnements intérieurs. Les robots marchants et roulants permettent de compléter deux tâches bien distinctes : s'affranchir des obstacles et des terrains complexes et atteindre des longues distances rapidement. Les travaux de [125] présentent un prototype de robot (ANYmal) équipé de roues non-directrices associées aux fins d'effecteurs des pattes, Figure 2.5 (B). Lors des expérimentations, le robot a pu démontrer sa capacité de franchissement lors du "Urban Circuit of the Subterranean Challenge", une compétition incluant des escaliers et obstacles dans un environnement sous-terrain. La vitesse moyenne du prototype a été 0,5 m/s, soit le double de vitesse pour un robot à patte classique. L'architecture du robot est complexe et nécessite un double contrôle de la locomotion : roues et pattes. La ressource calculatoire et les besoins électriques du système sont des contraintes de ce type de robot hybride. L'optimisation du temps de marche et l'hybridation marche-conduite font partie des futurs travaux de [125].

La cinquième catégorie : Les robots marchants.

Pour ce qui est des robots marchants, ils représentent parfaitement une approche à forte influence biomimétique. La démarche est généralement résolue par la cinématique directe, inverse et la matrice Denavit-Hartenberg permettant de modéliser les jointures. Le but étant de définir la localisation des fins d'effecteurs dans un repère cartésien ou polaire. Parmi eux, il existe différents systèmes dont une configuration à deux pattes (Walk man, [40]), quatre pattes (AQuRo [41]), six pattes (Weaver [43]), huit pattes (Spider-Robot [44]). Plusieurs combinaisons sont possibles : l'inspiration biomimétique des pattes (mammifère, arachnéen, reptiles) ou d'inspiration non-zoomorphe, « sous actionnés », patte télescopique, pattes hybrides. Ces systèmes marchants ont l'énorme avantage d'une

manœuvrabilité importante, d'une capacité de franchissement d'obstacles et de pente importante, tout en étant d'une faible interaction avec le sol et en restant disponibles en termes de niveau technologique. Néanmoins, ils sont gourmands en énergie et sont d'une importante complexité mécanique comparée aux autres modes de locomotion.

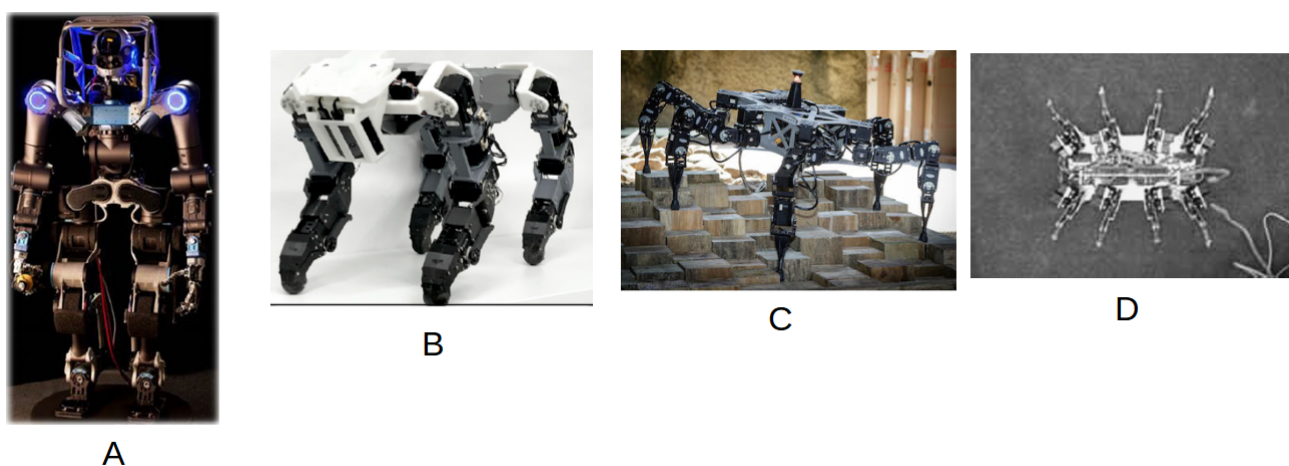


FIGURE 2.6 – A : Walk man (2 pattes) [40], B : AQuRo (4 pattes) [41], C : Weaver (6 pattes) [43], D : Spider-Robot (8 pattes) [44].

Dans cette partie, nous présentons les différents systèmes marchants issue de la Figure 2.6. Les architectures sont décrites par leurs cas d'usage et le bilan des degrés de liberté (pour leur dextérité). Walk man [40] est un robot humanoïde destiné à la manipulation et à l'utilisation d'outils dans les situations d'urgence. L'approche d'humanoïde présenté à travers Walk man, met en lumière 2 principales contributions. Walk man embarque un ensemble technologique permettant d'opérer dans un environnement déstructuré. Les contributions concernent des effecteurs puissants combinés à des capteurs proprioceptifs et un contrôle actif des impédances pour une meilleure adaptation, interaction et résistance mécanique. Un cadre de travail intégré comprenant la prise en charge de la locomotion et la manipulation permet le développement de comportements primitifs en matière locomanipulation et perception (saisir des objets, utiliser des outils, porter des charges, etc.). Les robots humanoïdes sont des architectures intéressantes pour les cas d'inspections complexes dans notre périmètre d'étude. Les principaux points forts de ces modèles de robot sont les actions possibles nécessitant de la dextérité (ouvrir une porte par exemple). Les axes d'amélioration sont le coût important des servomoteurs, le poids, l'encombrement, la tolérance à la faute moindre (2 jambes donc pas de redondance) et la capacité à se déplacer dans des endroits confinés et déstructurés.

Les robots avec des pattes ont l'avantage sur des terrains difficiles et sont plus adaptés pour des domaines d'application militaire et des contextes de catastrophe (séismes, effondrements, etc.). Le robot AQUaRo [41] est un robot à quatre pattes bio-inspiré des félins. Ce modèle embarque des capteurs de pression à l'extrémité des fins d'effecteur (patte) et offre de nouvelles capacités physiques et cognitifs. L'approche proposée par l'auteur inclut une construction d'une carte topologique à densité dynamique avec modèle d'attention, perception accessible à l'aide de la carte topologique et un modèle de locomotion basé sur un système d'apprentissage automatique par réseau de neurones. Les robots à 4 pattes inspirés de vrais animaux vivants sont légion dans la littérature. L'approche de l'auteur propose de s'inspirer du vivant pour résoudre des problèmes d'accessibilités de terrain en adoptant un design morphologique, notamment sur le design de la patte. Les capteurs proprioceptifs (IMU, capteur de pression) et extéroceptifs (laser 3D) couplés à l'architecture du robot contribuent à naviguer dans des terrains complexes pour un coût matériel de prototype d'environ 11 000 €.

Comme exemple de robot hexapode, à 6 pattes, nous pouvons citer Weaver [43]. Développé pour la navigation de milieux les plus complexes, le robot est équipé d'un IMU pour contrôler son inclinaison et d'un générateur de démarche basé sur la cinématique directe, inverse et du Jacobien. L'approche proposée par l'auteur fournit un contrôle de robots marchants permettant l'atteinte de milieux complexes et dégradés. Le prototype présenté possède de nombreux effecteurs pour adapter la position de l'hexapode à son environnement, et donc une meilleure stabilité. Ce type d'architecture est propice pour nos cas d'études, car il existe une redondance mécanique sur ce type de robot : 6 pattes. L'autonomie de la batterie et l'estimation de la rugosité sont des sujets à explorer, afin de réduire la sollicitation des servomoteurs en énergie et en efforts mécaniques.

Le robot à 8 pattes Spider-Robot [44] est un robot expérimental adapté à de multiples environnements. L'approche déployée par l'auteur se focalise sur le contrôle par logique floue ("fuzzy logic") et les réseaux de neurones pour réaliser un système de démarche, permettant une accélération progressive et une marche en mode "économie d'énergie". Le robot présenté possède un fonctionnement mécanique et comportemental proche de l'araignée. Le postulat de l'auteur est d'utiliser un maximum d'informations sur le monde extérieur et sur le robot araignée (calcul de la fatigue) conjointement pour réaliser un système de contrôle d'inspiration biologique.

Ce type de robot est adapté pour nos cas d'étude (cas 1, cas 2 et cas 3) pour la redondance

mécanique, la capacité d'emport et les capacités de navigation en milieux dégradés, notamment le franchissement d'obstacles. Néanmoins, l'énergie déployée pour la démarche est plus importante que pour un robot hexapode (avec 2 pattes en moins). Les travaux de l'auteur dressent une approche appréhendant ce problème d'énergie.

2.2.3 Synthèse générale : Catégorie de robot

Nous avons ici présenté les modèles illustrant les différentes catégories de robots terrestres. Suite à ces différentes présentations, nous allons réaliser le bilan et l'évaluation pour déterminer l'architecture la plus adaptée pour un environnement industriel en fonction des critères choisis. Nous résumons par la suite les résultats des différentes catégories : les robots à roues, à chenilles, sauteurs, hybrides et marchants. Les robots à roues possèdent une excellente capacité à atteindre une vitesse de déplacement maximale avec un bon ratio de consommation. En revanche, la capacité de franchir des obstacles et d'évoluer dans des environnements non-structurés est problématique pour les environnements industriels et chaotiques. Les robots à chenilles ont une bonne capacité à franchir les obstacles et à atteindre une vitesse de déplacement raisonnable. Ce type d'architecture souffre des mêmes problématiques que les robots à roues, leurs interactions avec le sol les limitent pour les terrains complexes. Et de ce fait, le franchissement d'obstacles se fait souvent en force. Les robots sauteurs excellent dans le déplacement quasi-instantané et le franchissement d'obstacle. Le point faible de ce type d'architecture est la manœuvrabilité et la tolérance à la faute. Même si des protocoles de récupération suite à une chute ont été démontrés dans la littérature, le calcul de la trajectoire et du point d'arrivée après le saut reste un sujet ouvert. Cette architecture ne serait pas adaptée pour un environnement industriel et chaotique. Les robots hybrides présentent des caractéristiques intéressantes par la fusion de 2 technologies de locomotion. Les robots roulants et marchants permettent par exemple, de franchir des obstacles en utilisant la marche et d'atteindre une vitesse optimale en mode roulant. La consommation d'énergie et l'accès à cette technologie sont à nuancer. La duplication des systèmes implique une consommation d'énergie plus importante des parties électroniques et motrices (servomoteurs et moteurs de roues), ainsi qu'un développement logiciel de contrôle et odométrique sur deux architectures. La complexité mécanique est elle aussi accrue. Les robots sauteurs et roulants présentent une stratégie différente pour franchir les obstacles et atteindre une vitesse de déplacement optimale. La locomotion est alors constante même en cas d'obstacle (le robot peut s'armer pour le saut tout en roulant.). En revanche, l'interaction avec le sol et le contrôle des trajectoires pour

le saut restent des points bloquants pour les environnements que l'on souhaite traiter en inspection. Les robots marchants présentent des capacités intéressantes. Le franchissement d'obstacles, la manœuvrabilité, la tolérance à la faute et leur faible interaction avec le sol, en font des robots de choix pour le domaine des inspections en environnements complexes et non structuré. Le ratio de consommation d'énergie par rapport au déplacement effectué est, certes, moins intéressant que sur des robots à déplacement continu. Le niveau de technologie disponible reste acceptable.

L'évaluation des différentes architectures

L'évaluation nous montre que l'architecture des robots marchants est la plus adaptée à notre cas. Ce moyen de locomotion permet au robot d'avoir une grande adaptabilité pour les terrains accidentés, le franchissement d'obstacles et la capacité de franchir des pentes, malgré leur mécanique et leur contrôle de la marche qui restent complexe.

Description	Roues	Chenilles	Marchants	Sauteurs	Roulant-Marchant hybride	Sauteur-Roulant hybride
Capacité de translation et rotation au même moment.	2	3	5	1	4	4
Tolérance à la faute	2	3	5	1	4	4
Capacité de se déplacer sur des terrains plats à une vitesse de 40 cm/s.	4	3	3	5	5	5
Capacité d'escalade progressive qui équivaut à la taille de l'élément mobile. (Diamètre de la roue, longueur des tibias et fémur.)	3	4	5	5	4	5
Capacité de montée en pente d'une inclinaison de 30°.	2	3	4	5	5	5
Dissipation de l'élément mobile pendant le mouvement sur le sol.	2	3	3	5	5	5
Niveau de complexité mécanique.	4	4	1	2	1	1
Capacité à exécuter et à poursuivre le mouvement après dysfonctionnement des organes de mobilités.	4	1	4	1	5	1
Capacité de déplacement avec une consommation de batterie moindre par distance unitaire.	4	3	2	4	1	5
Capacité d'emport compris en 8 à 20 % du poids à vide.	4	4	3	4	2	4
Influence du contact avec le sol pendant la locomotion / Interactions avec le sol.	3	2	4	2	3	1
Maturité de la technologie disponible.	5	3	4	2	2	2
Total	39	36	43	37	41	42

TABLE 2.2 – Evaluation des différentes architectures de robots mobiles suivant les critères définis précédemment.

Dans cette étude, nous avons remarqué que les robots hexapodes se différenciaient des autres architectures marchantes (bipèdes, quadrupède) par la stabilité lors de la marche et le nombre de démarches possible plus important. Les allures sont différentes car les deux cités sont des modèles basés sur des mammifères. Ils peuvent donc plus naturellement réalisés des foulées et des sprints.

2.3 État de l'art : les robots hexapodes

Nous avons vu précédemment que les robots marchants présentent des critères intéressants permettant de pouvoir évoluer dans les environnements industriels, à risques ou non

explorés par l'Homme. La problématique est de pouvoir déterminer une architecture de robot hexapode qui serait la plus adaptée aux besoins des différents cas d'inspection à réaliser : les zones de stockages d'hydrocarbure, les milieux navals, les carrières souterraines et les zones accidentées ou post-catastrophe. Dans cette section, nous allons évoquer un axe spécifique des robots marchants : les robots à 6 pattes, les hexapodes. Ces architectures issues du biomimétisme sont le fruit de recherches et d'analyses du monde des insectes. Dans cette section, nous aborderons les configurations possibles, les particularités et les différents robots issus de la littérature.

2.3.1 Introduction

Les robots marchants sont couramment utilisés par les militaires pour transporter des marchandises sur un terrain accidenté [54], pour l'inspection des centrales nucléaires [55] et pour les missions de recherche et de sauvetage [56]. En outre, certains robots à pattes sont développés pour l'exploration et la cartographie d'environnements inconnus [57], les missions d'inspection [58] et les missions de reconnaissance et de sauvetage [60]. Les robots marchants sont classés en fonction de leurs types de pattes, de leurs orientations et de leurs configurations articulaires. L'hexapode a été très souvent utilisé dans les milieux complexes, industriels, non explorés ou dangereux à explorer. L'inspection de milieux industriels soulève des problématiques de franchissement d'obstacle, de tolérance à la faute, de capacité d'emport pour réaliser l'analyse de l'environnement. L'utilisation de robots hexapodes offrent des avantages sur la locomotion. Ils sont souvent choisis pour tester des théories biologiques sur la locomotion des insectes, sur le contrôle de leur pattes et sur des aspects neuro-biologiques. Les hexapodes constituent le sous-embranchement le plus représenté en nombre d'espèces du règne animal. Ils comprennent tous les insectes ainsi que d'autres classes comme les protoures, les diploures et les collemboles [45]. Ces arthropodes se caractérisent tous par la présence de trois paires de pattes, donc six pattes, ce qui leur vaut le nom d'hexapode. Les insectes ont été choisis comme modèles parce que leur système nerveux est plus simple que celui d'autres espèces animales [46]. De plus, des comportements complexes peuvent être attribués à quelques neurones seulement et le chemin entre les entrées sensorielles et les sorties motrices est relativement court. L'architecture neuronale d'un insecte est un sujet actif de recherche pour leurs capacités biochimiques. Les archéognathes sont capables de détecter des molécules volatiles à l'aide de leur olfaction, récepteurs ionotroques et dans certains cas, avec leurs récepteurs gustatifs [48]. Les biologistes utilisent des robots hexapodes pour tester différentes hypothèses

sur la locomotion [47].

L'étude des hexapodes

Les robots hexapodes, principalement d'inspiration biologique, dépendent largement de l'espèce d'insecte utilisée comme modèle. Le cafard et le phasme sont les deux insectes les plus couramment utilisés [49], [50]. Ces deux insectes ont été étudiés de façon extensive sur le plan d'éthologie et sur celui de la neurophysiologie. Au moment présent, aucun système nerveux complet n'est connu, et donc, les modèles de robot s'inspirent habituellement des cafards, fourmis ou encore araignées, incluant éventuellement d'autres espèces encore [51], [52].

La locomotion

L'étude de la marche des insectes est d'un intérêt particulier pour présenter une alternative à l'usage de roues et ces limitations sur les terrains dégradés ou accidentés. Ce domaine, la robotique insectoïde, se réfère donc aux robots d'inspiration biologique imitant dans le cas présent les hexapodes. Les architectures à six pattes sont considérées plus stables que les bipèdes grâce à leur nombre supérieur de pattes : ils sont donc statiquement plus stables [53]. De ce fait, ils ne dépendent pas de contrôleurs en temps réel pour rester debout ou pour marcher. La démarche de l'hexapode dépend principalement du design du corps et des pattes.

En effet, ils possèdent des mouvements omnidirectionnels et discontinus, une tolérance à la « faute », une bonne stabilité, une importante variété de terrains accessibles, une capacité de franchissement d'obstacles plus important à taille équivalente (par rapport aux robots à roues, tractés...), un impact moindre sur le terrain et une plus grande mobilité sur terrain naturel. Ce type de robot est plus complexe à concevoir, développer et paramétrer à cause des degrés de liberté (minimum 18 DoF) [61]. Le coût est plus important dû à l'achat des servomoteurs (correspond à 80 % du coût matériel total d'un prototype). Il a une efficacité énergétique moindre, car beaucoup d'efforts sont engagés pour le déplacement et une vitesse moindre comparée à un robot à roues. Il y a plus de paramètres à prendre en compte dans le design des pattes et du corps [61]. Les efforts pour le déplacement holonome est plus important qu'avec un robot à roue holonome.

Les hexapodes peuvent avoir une odométrie calculée [62] par le contrôleur, à partir des calculs cinématiques, de la démarche et la vitesse de la démarche. Mais cela reste une

valeur "estimée" et beaucoup de paramètres peuvent influencer la précision (le pas des servomoteurs, le cycle de marche, la démarche, les capteurs, l'algorithme). Pour cette raison, ce type de robot doit avoir une source odométrique fiable afin d'assurer une localisation robuste.

2.3.2 Architecture des hexapodes

Depuis les années 1950 où les premières ébauches de robots hexapodes ont fait leur apparition, les recherches sur ces architectures de robots n'ont cessé de progresser. Les premiers prototypes avaient un système de commande simple : les séquences étaient pré-enregistrées. Ils ont depuis évolués vers des systèmes appréhendant les environnements difficiles en alliant un système de démarche, un dispositif de vision et une architecture décisionnelle [61]. Dans cette partie, nous allons aborder les différents aspects de l'architecture mécanique d'un robot hexapode en décomposant les différentes parties et en illustrant les solutions proposées dans la littérature. Les premiers hexapodes peuvent être identifiés comme des robots basés sur un mouvement totalement prédéterminé et ne pouvaient donc pas être posés au sol. Comme nous l'avons évoqué, il existe différentes architectures de robots marchants à configuration insectoïde ou mammifère par exemple. Ces modélisations s'appliquent aussi aux robots hexapodes [61].

L'influence du design sur la mobilité

Les facteurs clé à prendre en compte [61] sont les démarches, la vitesse, l'évitement d'obstacle, le franchissement d'obstacles, le poids embarquable, l'autonomie, les caractéristiques de mission et le coût. Par ailleurs, à cela, nous pouvons ajouter les caractéristiques environnementales (Voir Chapitre 2 - Analyse des éléments contextuels) qui peuvent influencer la stratégie de locomotion à adopter, et de ce fait, l'architecture du robot. Ces données d'entrées permettent de déterminer un design préliminaire et de définir les différentes caractéristiques. Parmi ces données de sorties, nous pouvons citer la structure mécanique, l'architecture des pattes, la taille maximale, les effecteurs, le mécanisme de contrôle et la capacité d'alimentation.

Les différentes configurations

Les travaux de [61] montres différentes configurations possibles. Les pattes par exemple, peuvent être bioinspirées (mammifères, arachnéens, reptiliennes) ou d'une autre forme non

bioinspirée (sous actionnées, télescopiques, hybrides). L'orientation des pattes influence le sens de la marche du robot (frontale, sagittale, circulaire) et l'orientation des pattes influence aussi la stabilité et la répartition du couple sur les différentes articulations (genoux orientés vers l'extérieur, l'intérieur, ou dans la même orientation). Ces possibilités citées sont non-exhaustives et les arrangements des effecteurs peuvent aussi impacter sur les capacités du robot.

Les exemples d'architecture de robot hexapode Comme nous l'avons vu précédemment, les robots à 6 pattes ou hexapodes, possèdent des avantages pour nos cas d'études, notamment la capacité de franchissement d'obstacles. Dans cette partie, nous verrons les différentes architectures d'hexapodes issues de la littérature.

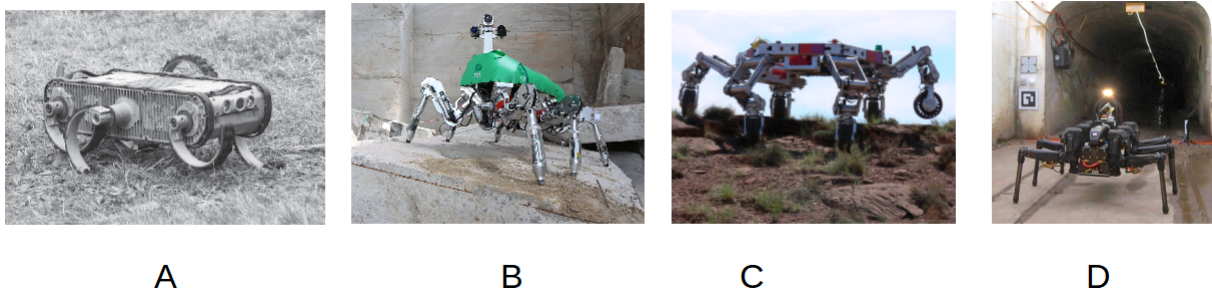


FIGURE 2.7 – Différents types d'architectures d'hexapode : (A) architecture de base RHex [63] ; (B) Le robot marcheur à six pattes LAURON V [66] ; (C) ATHLÈTE pendant les tests expérimentaux [68] ; (D) Bruce, un hexapode réalisé par l'université de Zurich [70].

La conception RHex [63] provient d'un effort multi-disciplinaire et multi-universitaire financé par la DARPA qui applique des techniques mathématiques de la théorie des systèmes dynamiques aux problèmes de locomotion animale. Les travaux de [63] ont mis au jour un prototype de robot hexapode possédant des demi-cercles comme patte. La conception de l'hexapode se compose d'un corps rigide avec six pattes souples, chacune avec un DoF (Figure 2.7). Ainsi, RHex n'a que six moteurs qui font tourner les jambes comme une roue. RHex a une architecture simple, permettant d'avoir un compromis entre vitesse et franchissement d'obstacle. Malgré son appellation d'hexapode, le robot n'est pas holonome et est donc limité dans des déplacements semi-continus. Il possède les avantages et les inconvénients d'un robot à roues. Cette architecture peut avoir des difficultés sur les obstacles qui sont supérieurs à son diamètre de roue. L'absence d'ordinateur embarqué ou de système odométrique et de navigation, limite l'usage dans des cas complexes

environnementaux.

L'hexapode Lauron [66] a été le fruit de travaux de recherche menés par le centre de recherche de technologie de l'information de Karlsruhe en Allemagne en partenariat avec le centre technologique de Poznan en Pologne. Le robot hexapode Lauron V (figure 2.7) est le résultat d'environ 10 ans d'amélioration progressive par rapport aux configurations précédentes Lauron I, II, III et IV. LAURON V [66] est biologiquement inspiré du phasme. Comme cet insecte, le robot a six pattes fixées à un corps central. Chacune des six pattes est actionnée par quatre articulations. Chaque pied a un capteur de force à trois axes et chaque moteur a un capteur de courant qui détecte les forces opposées à son mouvement. Le robot est équipé d'un PC embarqué et de capteurs infrarouges, caméra stéréoscopique, d'un laser multibandes et d'un laser rotatif 2D. Il peut donc modéliser et cartographier son environnement avec une précision importante. Il possède 2 heures d'autonomie. L'expérimentation de Lauron V a été produite lors d'un exercice EU Taranis [67]. Ce type d'exercice permet d'évaluer l'organisation, les compétences et l'efficacité d'une procédure de sauvetage dans des cas réels. Lauron V a démontré des capacités dans des scénarios complexes et en lien avec notre problématique d'inspection industrielle et sauvetage dans des environnements complexes avec une composante chimique. Son grand nombre d'effecteurs (26 pour les pattes et 2 pour la tête) nécessite un emport de batterie important et un poids accru (2kg par batterie 22,2V LiPo). Son autonomie de 2 heures maximum peut permettre de réaliser les principales tâches de reconnaissance pour un robot de 40 kg avec un poids embarquable de 10 kg maximum.

Destiné pour l'exploration spatiale, un robot lunaire à six pattes appelé ATHLETE [68] a été développé par le "Jet Propulsion Laboratory", NASA. Ce robot a la capacité de rouler rapidement sur des roues rotatives en un terrain plat et lisse et de marcher prudemment sur des roues fixes sur un terrain irrégulier et escarpé. ATHLÈTE avait une capacité de charge utile de 450 kg, un diamètre d'environ 4 m et une portée d'environ 6 m (figure 2.7). Chaque patte possède 6 DoF et une roue en fin d'effecteur pour un total de 36 DoF. ATHLETE [69] a la capacité de marcher et de rouler. Il peut franchir les terrains difficiles. ATHLETE est un robot expérimental encore en développement qui décompose la planification de trajectoire sur plusieurs couches. Néanmoins, ce type de prototype n'est pas encore mature et l'utilisation des roues dans nos cas d'inspection serait inadaptée, étant donné que la majorité des cas et des trajets se font sur des terrains complexes ou avec

du franchissement d'obstacles. L'opérateur peut le faire naviguer sur les différents points d'intérêt. Le robot a été testé par les opérateurs sur le mouvement et la navigation, mais l'implémentation de ce système n'est pas complet. Le prototype a été testé en simulation. La difficulté réside dans l'appréhension de deux différents espaces : l'espace de travail et la configuration de l'espace de marche. L'espace de travail fait référence, en l'occurrence, à son utilisation dans un environnement comportant des points de navigation, des terrains spécifiques et des obstacles [69]. La configuration de l'espace de marche peut impacter la locomotion du robot, sur un point de navigation où le chemin est inatteignable (obstacle par exemple) ou passe trop proche d'une singularité. La singularité d'une locomotion est due à un alignement particulier des joints, proche d'une vitesse infinie qui peut être requis dans une certaine direction sur les fins d'effecteurs.

L'hexapode Bruce [70] a été développé par le centre de recherche australien CSIRO et partiellement financé par le gouvernement américain dans le cadre du DARPA Subterranean Challenge. Les travaux de [70] ont présenté un robot hexapode imprimé à 80 % en impression 3D. Le robot peut traverser différents terrains comme les sols avec des végétaux, des rochers et des décombres à 0.3 m/s et peut aller jusqu'à une vitesse de 1 mètre par seconde. Il mesure 1 m de long et pèse 47 kg sans les batteries et peut emporter jusqu'à 3,2 kg de charge d'emport. Le robot Bruce présente les caractéristiques souhaitées pour nos cas d'usage. Il y a un bon ratio entre vitesse et franchissement d'obstacles. Le couple prévu pour ce prototype est important (80 Nm pour le coxis, 112 pour le fémur et 80 pour le tibia), ce qui rend le prototype couteux, uniquement sur les servomoteurs. Le robot a été expérimenté dans le DARPA SubT Challenge. Le mode de démarche tripode (3 pattes à la fois) a été utilisé à une vitesse 0.5 m/s sur 5 essais. Le robot a navigué sur 25 mètres sur un terrain composé de 25 m de pierre, de boue, et de pelouse. Un autre environnement d'essai, lors du DARPA SubT Challenge à Pittsburgh, a été utilisé pour démontrer la capacité du robot à marcher sur 15 m dans un environnement minier à une vitesse de 0.283 m/s, avant une défaillance du tibia arrière gauche. Ces expérimentations ont démontrées la capacité du robot à naviguer à 0.5 m/s sur un terrain complexe et accidenté.

2.3.3 Synthèse générale : Le choix d'architecture

Rhex est la fusion entre un robot marcheur et roulant : il est constitué de roues en demi-lune. La rotation de ces roues entraîne un effet de marche cyclique. C'est une architecture qui est limitée face à des obstacles dépassant le diamètre théorique de sa roue

complète. Le robot Lauron est un prototype possédant de nombreux degrés de liberté sur ces pattes et impliquent un coût d'achat important de servomoteurs. De plus, le but de cette recherche n'est pas de manipuler des objets, mais d'atteindre des sources de percepts. Le robot ATHLETE est constitué d'une architecture hybride avec la même problématique que Lauron : le nombre de servomoteurs. De plus, les milieux contraints et dégradés sont limités en accès pour les robots à roues et un robot de plus petite envergure est nécessaire (par exemple : cas 2 et cas 3). Bruce est une architecture polyvalente de type hexapode qui a démontré son utilité dans des environnements comparables à nos cas d'inspection. L'architecture présentée est une solution adaptée pour notre étude. Notre choix d'architecture d'hexapode s'est porté sur des pattes bio-inspirées de configuration arachnéenne orientée de manière frontale avec les genoux orientés vers les extérieurs. Les travaux de [61] portent sur les problèmes de conception des robots marchants. Dans cette étude, il montre une méthode d'analyse permettant d'appréhender au mieux le choix d'une architecture hexapode. Cet hexapode accueillera les capteurs pour les procédés de détection de points d'intérêts visuels et chimiques.

2.4 État de l'art : la détection des points d'intérêts visuels

Notre problématique réside dans la détection d'anomalies visuelles (fissure) et chimiques (gaz de type hydrocarbure). Ce problème d'ordre bimodal, comporte en effet une probabilité de détection de fissure et détection de fuite de gaz. Nos points d'intérêts sont donc visuels et chimiques dans un environnement industriel. Cependant, déterminer que l'image comporte un objet fissure ou qu'il y a une présence de gaz de type d'hydrocarbure ne suffit pas pour répondre à la problématique. À la question "quoi", nous devons ajouter la composante "où", c'est-à-dire localiser l'objet dans l'image et la source de la fuite de gaz. Ces deux approches sont respectivement l'asservissement visuel et chimique. Dans cette partie, nous allons nous intéresser aux algorithmes permettant de réaliser la détection des POI à travers un flux de données, que ce soit sur des images ou sur une série de données temporelle. Les principales problématiques concernent les bruits dans les images qui peuvent altérer la détection des défauts visuels (fissures, corrosion, défaut de soudure) et les flux d'air qui peuvent aussi perturber la recherche d'une source de gaz.

Les méthodes de détection d'anomalies visuelles

La détection d'objet ou reconnaissance d'objet est une tâche importante dans le domaine de la vision par ordinateur. À travers des images numériques composées de colonnes et de lignes de pixels (avec une variation RGB, "Red-Green-Blue" ou Rouge-Bleu-Vert), chaque pixel ou groupe de pixels est analysé pour déterminer les caractéristiques de l'objet. Ces caractéristiques servent de discriminants et/ou de base d'apprentissage pour entraîner un modèle permettant de reconnaître l'objet dans l'image.

La reconnaissance de fissures est un sujet de recherche actif dans le monde de l'industrie, mais aussi dans d'autres domaines, telle que le domaine militaire ou médical. Il a beaucoup évolué durant les 20 dernières années. Les principales problématiques sur ce type d'opération réalisé en manuel sont généralement le coût financier, l'expertise nécessaire, le temps, les outils de mesures et la marge d'erreur. L'intérêt est d'engager des méthodes autonomes de détection et d'avoir un bilan probabiliste sur la détection de l'objet. Dans cette partie, nous allons évoquer les différents type de méthodes permettant de réaliser la détection de fissure et de les évaluer suivants les critères de précision, de temps de calcul, de robustesse photométrique et géométrique, ainsi que la puissance de calcul nécessaire.

2.4.1 Les catégories de méthodes

La reconnaissance ou détection d'objet dans l'image se manifeste par la présence d'une boîte englobante entourant le ou les objets détectés. Il existe aussi des formes hybrides où par exemple, la reconnaissance ou détection d'objet va permettre aussi de pouvoir compter les objets.

Les problématiques de la détection d'objet.

Les principales problématiques de la détection d'objets sont les suivantes [73] :

- Perturbations photométriques (changement d'intensité et couleur d'éclairage, ombres d'autres objets, conditions atmosphériques : brouillard, etc.).
- Perturbations géométriques : position (translation) et angle (rotation), taille (échelle, homothétie), changement de point de vue (approché par une transformation affine ou homographique), occultations, parties manquantes.
- Objets avec formes identiques, mais apparences différentes (couleurs, fonction, etc.).

- Objets empilés et pouvant être confondus en groupe au lieu d'être identifiés de manière individuelle.
- Base de données importante à réaliser pour représenter une diversité suffisante (1000 images minimum).

Les approches classiques ou traditionnelles sont généralement peu coûteuses d'un point de vue calculatoire. En revanche, elles sont très sensibles aux variations photométriques et géométriques. En effet, leur manque de capacité de généraliser l'objet en question limite son utilisation, mais elles peuvent être associées à une autre approche pour compléter les lacunes de celle-ci. Nous présenterons ici des méthodes visuelles, traditionnelles, basées sur un concept mathématique et/ou statistique par l'utilisation de transformée, de vectorisation, de gradient ou encore d'histogramme afin d'extraire des caractéristiques locales et détecter les objets, comme les défauts industriels (fissures).

Les techniques de corrélation [74] ont souvent été utilisées pour la détection d'objet comme les fissures. Pour chaque position, l'algorithme calcule la distance entre l'image modèle et la zone de l'image associée. Le modèle d'objet est cependant très restreint et le modèle est sujet à des perturbations photométriques et géométriques. La transformée de Hough [76], [75] permet de détecter des formes simples comme des lignes ou l'utilisation de la dualité ligne-point. Le paramétrage peut être de type cartésien, mais dans ce cas-là, il engendre des problèmes avec les verticales. En paramétrage polaire, le calcul est plus lourd, mais permet de s'affranchir des problèmes avec les lignes verticales.

Le premier algorithme ayant eu des résultats compétitifs est la technique des haar-cascade [94]. Les caractéristiques sont « codées dans le dur » (Haar Cascades) et ensuite entraîné à l'aide d'un algorithme de Support Vector Machine (SVM) pour classer l'ensemble de caractéristiques. La limite de cet algorithme provient du fait qu'il fut incapable de détecter les visages ayant une variation géométrique ou photométrique (luminosité, angle de prise de vue, distorsion, bruit). L'idée derrière ce concept consiste à décrire par l'intensité de distribution des gradients ou de vecteurs les apparences de l'objets local (caractéristiques de type HAAR, HOG, SIFT, SURF) [95]. Cet algorithme a du mal à généraliser et souffre du bruit de l'arrière-plan et d'un manque de précision. L'approche SIFT [77], est utile pour l'extraction de caractéristiques d'image. L'algorithme prend une image et la transforme en une grande collection de vecteurs de caractéristiques locales. Chacun de ces vecteurs de caractéristiques est invariant pour toute mise à l'échelle, rotation ou translation de l'image. L'approche SIFT connaît des limites. L'algorithme est

sensible au flou ou à l'estompage de contours, mais reste robuste sur des changements de position d'échelle (ratio entre 2 à 2,5) et de rotation (30 à 45°). Au-delà de 50° en rotation, les résultats sont incertains.

Les approches classiques sont très sensibles aux variations géométriques, photométriques ainsi que les angles de prises de vue. Ces approches sont devenues quasiment obsolètes avec les méthodes d'apprentissage automatique permettant de prendre en compte les différentes variations et bruits dans l'image dans une base de données.

Les méthodes visuelles d'apprentissage profond

Les méthodes d'apprentissage profond doivent résoudre 2 tâches principales. Pour la première tâche, leur mission est de trouver le nombre d'objets détectés. Puis pour la seconde tâche, ils doivent classer ses objets et estimé la taille de la boite englobante, qui servira à localiser l'objet détecté dans l'image. Certaines méthodes sont séquentielles. D'autres méthodes, parallélisent ces opérations pour les réaliser en 1 seule, et même étape. Les méthodes d'apprentissage par réseaux de neurones peuvent donc être séparées en 2 catégories. Les méthodes en 2 étapes et les méthodes en 1 étape.

Les méthodes visuelles d'apprentissage profond en 2 étapes

Les approches par apprentissage automatiques ont l'avantage de pouvoir créer une inférence représentative de la base de données et donc, de ce que l'on souhaite détecter. Les méthodes supervisées nécessitent une étape de labellisation ou de supervision humaine pour discriminer les classes au préalable. Les méthodes non supervisées n'utilisent pas l'étape de labellisation. La stratégie est d'explorer le jeu de données pour en extraire les caractéristiques clé permettant de discriminer automatiquement les différentes classes d'objets ou de tendre vers un comportement souhaité. Les approches supervisées par apprentissage automatique utilisent les réseaux de neurones convolutifs. Les techniques d'apprentissage en profondeur sont capables de détecter des objets de bout en bout sans définir spécifiquement les fonctionnalités et sont généralement basées sur des réseaux de neurones convolutifs (CNN) [78]. Un réseau neuronal convolutif (CNN ou ConvNet) est un type spécial de réseaux neuronaux multicouches, conçu pour reconnaître les points d'intérêts visuels (objets) directement à partir des pixels de l'image. La présentation de ces algorithmes se fait dans le cadre de leur utilisation dans un intérêt industriel, pour les cas de notre étude, le cas 1 pour les stockages d'hydrocarbure et environnement naval, le cas 2 pour les carrières souterraines et le cas 3 pour les environnements post-catastrophes.

En 2012, AlexNet [229] a nettement surpassé tous ses concurrents précédents au Défi de reconnaissance visuelle à grande échelle ImageNet (ILSVRC) et a remporté le défi. Les réseaux de neurones convolutifs sont devenus la référence en matière de classification d'images après la performance du CNN de Krizhevsky [229] pendant la compétition ImageNet. Les travaux de [80] établissent une étude comparative des algorithmes de détection de type CNN pour le décèlement des fissures dans les images. Les algorithmes comparés incluent AlexNet, VGG (16 et 19), GoogleNet et ResNet (50, 101 et 152) avec une base de données de 40000 images d'une dimension de 224 sur 224 pixels.

Les CNN (Convolutional Neural Network) étaient trop lents et très coûteux en termes de calcul. R-CNN résout ce problème en utilisant un algorithme de proposition d'objet appelé "recherche sélective" qui réduit le nombre de cadres de délimitation qui sont fournis au classificateur à près de 2000 propositions de région. Dans R-CNN, la méthode de recherche sélective [81] est une alternative à la recherche exhaustive dans une image pour capturer l'emplacement de l'objet. Il regarde l'image à travers des fenêtres de différentes tailles, et pour chaque taille essaie de regrouper les pixels adjacents par texture, couleur ou intensité pour identifier les objets. R-CNN a été amélioré au fil du temps pour de meilleures performances. Le réseau convolutionnel basé sur la région rapide (Fast R-CNN) [82] a réduit la consommation de temps liée au nombre élevé de modèles nécessaires pour analyser toutes les propositions de région dans R-CNN. Les travaux de Chen [71] ont démontré que l'utilisation d'un réseau de neurones convolutifs associé à une méthode de décision Bayésienne permet d'obtenir des résultats à 98,3% de précision avec un taux de faux positif de 0,1 %.

Les méthodes visuelles d'apprentissage profond directes

Afin de détecter les objets dans l'image (les fissures), nous utiliserons un algorithme d'apprentissage automatique supervisé. Les réseaux de neurones convolutifs permettent de détecter plusieurs classes d'objets tout en détectant l'objet dans l'image à l'aide d'une boîte englobante. Les méthodes visuelles d'apprentissage profond en une étape représentent la meilleure alternative pour identifier les fissures en temps réel. YOLO ou "You Only Look Once" [83] est un algorithme de détection d'objets très différent des algorithmes basés sur les régions vus ci-dessus. Dans YOLO, un seul réseau convolutif prédit les boîtes englobantes et les probabilités de classe pour ces boîtes. Le modèle YOLO prédit directement les boîtes englobantes et les probabilités de classe avec un seul réseau en une seule évaluation. Ils recadrent la détection d'objet comme un problème de régression unique,

directement des pixels de l'image aux coordonnées de la boîte englobante et aux probabilités de classe. YOLO divise chaque image en une grille de $S \times S$ et chaque grille prédit N boîtes englobantes et la confiance. Le score de confiance nous indique à quel point il est certain que la boîte englobante prévue renferme en fait un objet. Les travaux de [84] utilisent YOLO (version 3) afin de détecter des fissures dans la chaussée, un environnement proche des cas que l'on peut rencontrer dans le domaine industriel. La métrique utilisée est la précision moyenne (mAP pour "Mean Average Précision"). Cette métrique est utilisée pour évaluer le modèle de détection d'objet compris entre 0 et 1 et est basé sur d'autres mesures : Matrice de confusion, Union sur l'intersection (IoU), le rappel et la précision. La précision sur la base de test (400 images) démontre un score de 51.2 % pour le mAP et 88 % en précision avec 60 FPS ("frame per second"). La métrique présentée démontre des performances correctes en précision et en traitement en temps réel. En revanche, la précision de la boîte englobante est perfectible et il y a un "certain taux" [84] de détection manqué à cause de la diversité des fissures et d'une possible pollution visuelle liée à l'environnement et aux conditions météorologiques. Le modèle SSD [85] a été publié, peu de temps après le modèle YOLO, et a également été affiné plus tard dans un article ultérieur. Contrairement à YOLO, SSD ne divise pas l'image en grilles de taille arbitraire, mais prédit le décalage des boîtes d'ancrage prédéfinies pour chaque emplacement de la carte des caractéristiques. Chaque boîte a une taille et une position fixes par rapport à sa cellule correspondante. Toutes les boîtes d'ancrage recouvrent l'ensemble de la carte des caractéristiques de manière convolutive. Les cartes d'entités à différents niveaux ont différentes tailles de champ réceptif. Les cases d'ancrage des différents niveaux sont redimensionnées de sorte qu'une carte d'entités n'est responsable que des objets avec une échelle particulière. Les travaux de [86] proposent une approche utilisant un drone et l'algorithme SSD pour détecter la condition des infrastructures, notamment les fissures. La méthode propose une notation de 0 à 9 décrivant l'état structurel couplé à un ensemble de 15 défauts structurels. Le but de l'approche est de proposer une carte 3D de l'état structurel des infrastructures. Afin de constituer la base de données, une base d'images originales, filtrées et thermiques sont utilisées. L'algorithme atteint une précision d'environ 70 % sur la partie validation. L'auteur décrit différentes limites sur ce procédé : la base de données des fissures est limitée, les conditions environnementales peuvent détériorer la précision de la détection, le PC embarqué (Raspberry Pi) est limité en puissance calculatoire et une architecture avec GPU est nécessaire. Au lieu de cela, des parties de l'image qui ont de fortes probabilités de contenir l'objet sont analysées. Les performances des différents

algorithmes YOLO [83], YOLOv2 ou YOLO9000 [88], YOLOv3 [87], YOLOv4 [89], YOLOv5 [90] sont présentés, figure 2.8 (Redmon et al. 2017). Ces méthodes offrent de bons résultats, tant sur le temps de détection que sur la précision. Cependant, la règle à un objet limite la proximité des objets détectés. Pour cela, YOLO a certaines limitations sur la proximité des objets. Il demande beaucoup d'images et une annotation précise, et de la ressource calculatoire (GPU) pour faire tourner des réseaux denses et précis.

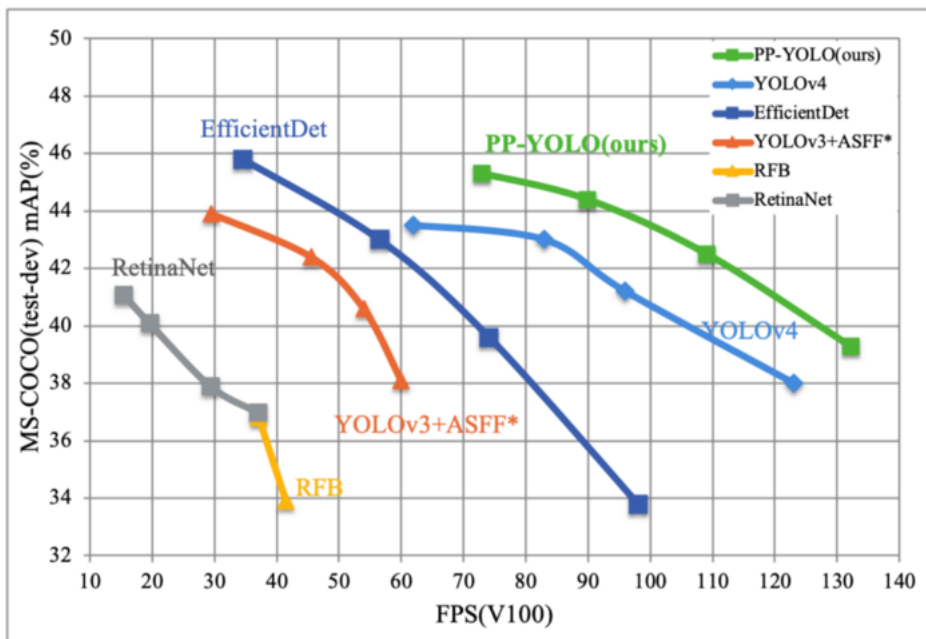


FIGURE 2.8 – Évaluation des récentes architectures de YOLO : YOLO4, YOLO5 et PP-YOLO [91]

Les algorithmes traditionnels présentent l'avantage d'avoir un coût calculatoire modéré comparé aux architectures embarquées proposées sur le marché. Ils restent cependant sensibles aux variations géométriques et photométriques. Ces critères sont importants dans nos cas et ont été révélés lors du Chapitre 2 - Analyse des éléments contextuels. Les algorithmes à 2 étapes et 1 étape, issues de l'apprentissage automatique supervisé par réseau de neurones, présentent les avantages d'être beaucoup moins sensibles aux variations photométriques et géométriques, de pouvoir détecter une large sélection de classes. En revanche, le coût calculatoire est très important par rapport à une architecture embarquée et nécessite une large base de données pour être viable sur le long terme et éviter le sur-apprentissage. Dans la prochaine et dernière partie sur la détection des points d'intérêts visuels, nous évaluerons les différents algorithmes par rapport à des

critères définis et aux contraintes évoquées lors du Chapitre 2 et celle de l'entreprise Pilgrim Technology, notamment en matière de retour d'expérience sur les opérations de photogrammétrie sur des ouvrages industriels.

2.4.2 Évaluation de l'architecture de détection des points d'intérêts visuels

Les critères utilisés sont classés en 3 niveaux (faible, moyen, fort) pour évaluer la précision des algorithmes de détection. Ces critères sont issues du tableau 2.9 en chapitre 2. Les critères sont la précision, le temps de calcul, la résilience aux bruits photométriques et géométriques et la puissance embarquée nécessaire. La précision nous permet de définir la robustesse de l'algorithme en situation réelle. La résilience aux bruits d'images définit la capacité de l'algorithme à pouvoir produire une détection malgré les nuisances pouvant apparaître sur le flux de la caméra. Enfin, la puissance embarquée dimensionne le besoin calculatoire, en considérant que la navigation autonome et le contrôleur de l'hexapode possèdent un coût de calcul. La matrice de cotation des différentes approches de détection visuelle est présentée dans la Figure 2.3.

Cotation	Précision	Temps de calcul	Photométrique	Géométrique	Puissance hardware (Profils embarquables)
1	Faible (25%)	Post Traitement Image par Image (>2FPS)	Très faible résistance (0%)	Très faible résistance (0%)	RPI 4 ou Odroid XU4 Architecture ARM Cortex A72 4Go RAM 4x1,50 GHz Pas de GPU ou GPU Mali De 1 à 32 GFlops
3	Moyenne (50%)	Latence Semi temps réel (>10FPS)	Moyenne (50%)	Moyenne (50%)	Latte Panda Alpha ou Jetson Nano ou Jetson TX2 Architecture ARM 8Go de RAM GPU Intel HD ou GPU Cuda De 246 GFLOPs à 1,26 TFLOPS
5	Forte (90%)	Temps réel (>24FPS)	Forte (100%)	Forte (100%)	Jetson NX ou Jetson AGX Xavier CPU 64bit 6 coeurs Addon hardware vision & IA 8Go à 12 Go de RAM GPU Volta et Tensor 21 TFLOPS à 100 TFLOPS

TABLE 2.3 – Matrice de cotation des différentes approches de détection visuelle.

Les architectures présentées précédemment sont évaluées. Nous retrouvons les approches classiques comme les techniques de corrélation, les transformés de Hough, les

algorithmes génétiques, ainsi que les approches modernes incluant de l'apprentissage profond supervisé comme les CNN, R-CNN, etc. Le résultat de ces différentes évaluations est présenté dans le tableau 2.4.

	Précision	Temps de calcul	Photométrique	Géométrique	Puissance Hardware
Technique de corrélation	1	1	1	1	1
Transformée de Hough	1	1	1	1	1
Transformée de Hough en polaire	1	1	1	1	1
Méthode de Hough généralisée	1	1	1	1	1
Méthode de Hough généralisée avec tangente	1	1	1	1	1
Méthode de Hough généralisée avec tangente	1	1	1	1	1
Les algorithmes génétiques	3	1	3	3	5
Les algorithmes d'apprentissage profond par renforcement.	3	5	5	5	5
Technique de haar-cascade	1	5	3	1	1
Les histogrammes orientés gradients pour les caractéristiques (HOG).	1	3	3	3	1
Transformation d'entité invariante d'échelle (SIFT)	3	3	3	3	1-3
Les réseaux de neurones convolutifs (CNN)	5	5	5	5	3-5
Les réseaux convolutif régional (R-CNN)	5	5	5	5	3-5

TABLE 2.4 – Tableau d'évaluation des architectures de détection visuel.

L'évaluation nous montre que les architectures modernes sont les plus adaptées pour nos différents cas d'application. Les réseaux de neurones convolutifs classiques (CNN) ou par région (R-CNN) obtiennent les scores les plus élevés.

2.4.3 Synthèse générale : choix de la méthode visuelle

Pour conclure, cette première analyse met en relief les problématiques de perturbation de détection liées à l'environnement (photométrique et géométrique). Nous choisissons l'algorithme YOLO pour effectuer la partie de détection automatique des points d'intérêts visuels pour sa portabilité sur des systèmes embarqués, son exécution en temps réel et sa précision.

2.5 État de l'art : la détection de points intérêts chimiques

L'asservissement visuel et chimique d'un robot sont deux tâches bien distinctes basées sur des types de données différentes, mais le but est le même : atteindre les points d'intérêts. Dans cette section, nous verrons les méthodes de captation chimique, de simulation et enfin d'asservissement.

2.5.1 Introduction

La problématique de détecter et mesurer les gaz industriels est une thématique importante. Différentes méthodes de mesures chimiques ont été développées au cours des dernières décennies pour traiter la cartographie chimique [159]. L'utilisation d'un système de caméra infrarouge [160] permet de détecter les fuites de méthane (CH_4), d'éthylène et de propylène avec une grande précision. L'imagerie Terahertz [161] appliquée à l'inspection non destructive et à l'analyse chimique, détecte les agents chimiques [162] en utilisant la chromatographie en phase gazeuse (GC) ou la spectrométrie de mobilité ionique (IMS). Ce type de technologie est coûteux, nécessite une puissance de calcul élevée, n'est souvent pas portable et est lent à traiter même s'il reste très précis et couvre de grandes zones (avec davantage de données à exploiter).

2.5.2 Les méthodes de simulation

En raison de la dangerosité des expérimentations utilisant du gaz réel, la simulation de dispersion de gaz (GDS) est une pratique appropriée pour anticiper les problèmes et améliorer la recherche et le développement en robotique. La simulation de dispersion de gaz doit traiter l'échelle spatiale [165], le modèle de dispersion [166] et la concentration de gaz [167]. Les modèles GDS simples peuvent être classés en : boîtes basées sur des modèles (modélisation de la concentration de gaz homogène dans des boîtes) [167], modèle gaussien et modèle analytique. Certaines approches avancées comme le modèle eulérien qui utilise un maillage cartésien 3D fixe comme référentiel à l'opposé des modèles de Lagrange, utilisant un référentiel mobile, permet d'obtenir des résultats plus précis. Le principal inconvénient est le coût de calcul élevé lorsque le nombre de particules augmente [168] [169]. Plusieurs implémentations d'outils de simulation de source de gaz sont présentées dans la littérature comme : Ansys [171] et OpenFoam [170]. Les deux solutions peuvent produire

un scénario complexe de modèle de dispersion, mais ne peuvent pas inclure en même temps une plate-forme robotique et une émulation de détection de gaz. Plumesim [172] a réussi à lancer une application de robotique avec une simulation de dispersion de gaz suivant le modèle de panaches gaussiens. Cependant, la théorie de simulation sous-jacente ne permet pas un scénario de gaz complexe ou une dispersion du panache. La figure 2.9 donne une comparaison des progiciels de distribution de sources de gaz pour les applications robotiques.

	Algorithmes originaux			Combinaisons d'algorithmes	
	Algorithme 1 : Système réactif basé sur les transitoires d'Ishida	Algorithme 2 : Estimation basée sur un modèle de moyenne temporelle.	Algorithme 3 : Infotaxis	Algorithme 4 : Estimation basée sur un modèle de moyenne temporelle : Prélèvement d'échantillons : Système réactif basé sur les transitoires d'Ishida	Algorithme 5 : Combinaison : Réactif dans une dimension, estimation basée sur un modèle de moyenne temporelle dans l'autre
Nombre d'essais	25	13	12	12	12/25
Distance entre la cible et la ligne de vent					
Distance moyenne en cm.	24,4	19,2	42,6 (a)	41,8	24,4
Taux de réussite : dans un rayon de 10 cm	32 %	0 %	8 %	8 %	32 %
Taux de réussite : dans un rayon de 25 cm	64 %	92 %	17 %	8 %	64 %
Taux de réussite : dans un rayon de 50 cm	92 %	100 %	42 %	75 %	92 %
Distance absolue par rapport à la source					
Distance moyenne en cm.		36	84,8 (a)	45,5	35,9
Taux de réussite : dans un rayon de 25 cm	Indisponible	31 %	0 %	8 %	25 %
Taux de réussite : dans un rayon de 50 cm	Indisponible	77 %	25 %	50 %	92 %
Taux de réussite : dans un rayon de 75 cm	Indisponible	100 %	33 %	100 %	100 %
Temps d'achèvement moyen (s)	163	760 (b) (c)	1243 (a)	Temps du système réactif transitoire d'Ishida + temps d'estimation	Temps du système réactif transitoire d'Ishida + temps d'estimation
Nombre moyen d'itérations	24,2	8	40,7 (a)	Identique au système réactif transitoire d'Ishida (c)	Identique au système réactif transitoire d'Ishida (c)
Possibilité d'adaptation à trois dimensions	Difficilement possible	Partiellement possible (d)	Facile	Difficilement possible	Difficilement possible

(a) Parmi les 7 points non considérés comme des échecs

(b) Non optimisé

(c) L'estimation ajoute quelque 1 à 2 secondes par itération

(d) Modèle beaucoup plus complexe nécessaire

FIGURE 2.9 – Comparaison des outils de simulation permettant la recherche de source de gaz pour une application robotique [176].

2.5.3 Asservissement chimique : La localisation des sources de gaz

La localisation des sources de gaz (GSL pour "Gas Source Localization") est le dernier processus clé pour permettre l'asservissement chimique, applicable pour des zones

intérieures du domaine industriel [172]. Certaines stratégies sont proposées dans la littérature : stratégie hybride utilisant l'algorithme de vision et d'olfaction (Player [173]), les stratégies de modèle moyennées dans le temps [182], et stratégie d'Infotaxis (OpenMORA [174]). Une autre façon de traiter GSL avec un faible besoin de calcul est le capteur de nez électronique ("e-noze") . Cette approche sensorielle permet d'utiliser des algorithmes GSL spécifiques et efficaces basés sur la localisation estimée de la source. La recherche itérative de polluant est la base de l'algorithme (SPIRAL) [175] qui vise à localiser une source de gaz dans un environnement intérieur dans l'absence d'un flux d'air fort et constant. D'autre part, l'algorithme de surtension ("surge cast") [183] a besoin d'un débit d'air constant et des données d'un anémomètre (direction du flux d'air) pour traiter le GSL. L'utilisation de la robotique mobile et d'algorithme GSL semble être un bon moyen d'obtenir un asservissement chimique.

2.5.4 L'asservissement chimique d'un robot hexapode

Certaines approches d'asservissement chimique, visent à rechercher le gaz carbonique [135] ou la direction du vent [191], à l'aide d'un robot hexapode. La localisation de sources de gaz dangereuses ou industrielles à l'aide de robots hexapodes ou de robots à pattes est encore une question de recherche ouverte. Les algorithmes de recherche de gaz interviennent dans la détection et dans l'atteinte de la source de gaz en question. Pour cela, les points d'intérêts nécessaires sont la concentration du gaz et la mesure du flux d'air. Ces cas-là sont valables lorsque le gaz est en état entropique. Nous présentons ici plusieurs algorithmes de recherches de gaz issus de la littérature.

L'algorithme d'écoulement ("Cast") [205] est un algorithme basé le flux d'air et sur un angle calculé en fonction de la distance de perte du panache, figure 2.10. C'est une méthode par itération où la perte du flux fait partie du process.

Un robot dans le panache remonte au vent avec un angle β (par rapport à la direction du vent mesuré) jusqu'à ce qu'il soit hors du panache pour une certaine distance, notée "dlost" pour la distance de perte. Une fois le panache perdu, le robot tourne et se déplace dans le sens du vent jusqu'à ce qu'il rencontre un paquet d'odeurs, puis remonte au vent avec l'angle β à nouveau. La direction du vent est mesurée à chaque fois que le robot passe en ré-acquisition du panache, et lorsqu'il rencontre le flux du gaz à nouveau. SPIRAL [206] est un algorithme permettant de localiser une source chimique sans utiliser une information de débit d'air. Ceci permet de pouvoir travailler de manière fiable dans un environnement intérieur sans avoir des flux d'air importants. L'algorithme SPIRAL

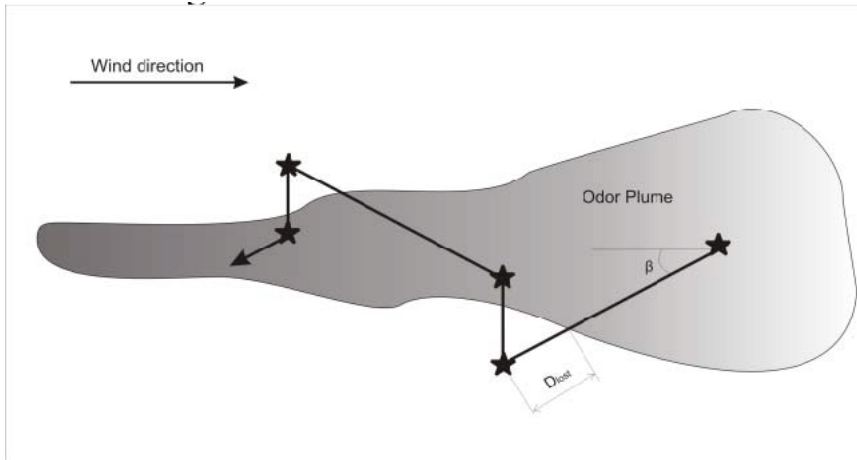


FIGURE 2.10 – Algorithme "Cast" [181].

suggère que le robot doit réaliser des mouvements en spirale pour retrouver la localisation de la source de gaz. Pendant le mouvement en spirale, le robot s'arrête pour prélever du gaz. Avec les données acquises, il détermine sa proximité avec la source : s'il récupère des informations du panache et qu'il est plus proche de la source par rapport aux mesures précédentes, le robot commence un autre mouvement en spirale, sinon, il continue dans le mouvement actuel.

L'algorithme de surtension-spirale ("surge-spiral") est similaire à l'algorithme de Hayes présenté [207], sauf que nous nous concentrons exclusivement sur son utilisation pour le suivi du panache ici. Par conséquent, nous avons un seul espace en paramètre. Un robot dans le panache se déplace droit vers le flux d'air jusqu'à ce qu'il perde le panache sur une "distance de perte", d_{lost} . Il essaie alors de réacquérir le panache en se déplaçant le long d'une spirale d'Archimède avec une taille d'écart d_{gap} . Contrairement à [207], nous commençons notre spirale dans le sens du vent. La direction du vent est mesurée lorsque le robot est en dehors du panache, et quand il revient dans le flux gazeux. Ce procédé est illustré figure 2.11.

L'algorithme de surtension ("surge-cast") [205] est une combinaison de houle et coulée par vent de travers. Il est similaire à l'algorithme "surge spirale", la spirale étant remplacée par le vent de travers mouvement, figure 2.12.

Un robot dans le panache se déplace droit vers le vent jusqu'à ce qu'il perde le panache sur une distance perdue. Il essaie alors de réacquérir le panache en déplaçant le vent de travers sur une distance définie (d_{cast}), d'abord d'un côté puis de l'autre. Pour maximiser

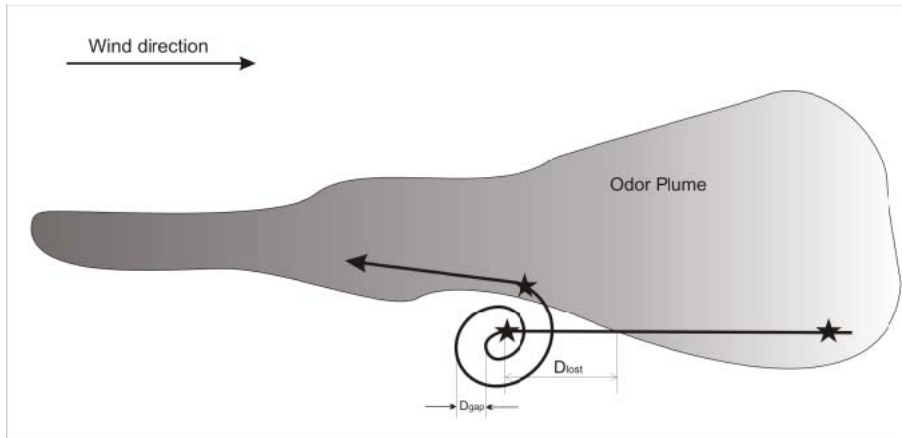


FIGURE 2.11 – Algorithme "surge-spiral" [207]

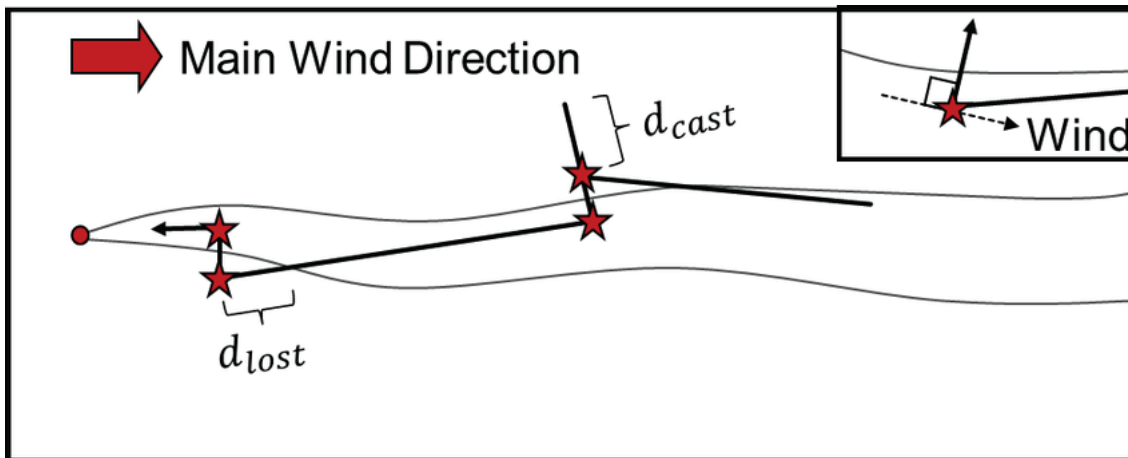


FIGURE 2.12 – Algorithme "surge-cast" [205]

les chances de heurter le panache lors du premier mouvement de vent de travers, le robot mesure la direction du vent pour estimer à partir du côté où il a perdu le panache. Si le robot n'a pas réacquis le panache, la course est considéré comme infructueux. Dans une application réelle, le robot reviendrait probablement au comportement de recherche de panache, ou essayerai de réacquérir le panache avec une plus grande distance d'approche ou avec une méthode en spirale. La direction du vent est mesurée lorsque le robot est hors du panache et quand il revient dans le flux d'air, comme indiqué figure 2.12, 2.10. Ces algorithmes possèdent le point commun d'utiliser le flux d'air et la concentration du gaz pour converger vers sa source. La précision et leur efficacité en environnement réaliste, intérieur et avec des obstacles se dégradent, dû à ces perturbations. L'utilisation d'un algorithme permettant de s'affranchir de ces limites est primordiale pour nos cas

d'étude. Les environnements industriels, les carrières souterraines et les milieux contraints et dégradés sont sujet à ce type de configuration et engendre des turbulences d'air ajoutant du bruit dans les mesures. Ce bruit impacte fortement la réalisation d'un asservissement chimique [208].

L'algorithme Grille est issu des recherches de Ojeda et al, [208]. Cet algorithme permet de pouvoir s'affranchir des limites évoquées précédemment. Cette approche probabiliste s'articule autour de la propagation d'estimations locales dans tout l'environnement. En exploitant la géométrie du milieu comme base de cette propagation, on évite de s'appuyer sur des modèles analytiques de dispersion, éliminant le besoin d'assumer un environnement contrôlé sous des conditions spécifiques. L'algorithme présenté dans ce travail suit la norme de structure d'un GSL basé sur l'estimation. Cette approche se déroule en 3 étapes. À partir d'un point précis emplacement, le robot (1) effectue une nouvelle mesure : gaz et vent dans notre cas (2), le robot met à jour l'estimation de l'état en fonction de leur concordance avec les modèles d'observation et (3) déterminent où se déplacer par la suite. Cette boucle continue jusqu'à ce que la source soit localisée et atteinte (déclaration de la source), qui est déterminé par un seuil de convergence. Lorsque le robot détecte un panache gazeux, un estimateur "local" quantifie pour chacune des cellules voisines la probabilité que le gaz l'ait traversée avant d'atteindre le robot cellule.

2.5.5 Synthèse générale : choix de la méthode chimique

Nous avons vu précédemment différents algorithmes de recherche de gaz. L'algorithme de recherche de gaz que nous souhaitons utiliser est l'algorithme Grille ("Grid"). Dans cette étude, cet algorithme sera utilisé et comparé aux méthodes mentionnées ci-dessus.

2.6 Conclusion de fin de chapitre

Nous avons vu précédemment l'importance de pouvoir évaluer l'environnement du robot afin de caractériser les différentes difficultés du milieu. La robotique mobile possède plusieurs architectures pour différentes applications. L'architecture choisie est donc un robot hexapode de type insectoïde. À la connaissance de l'auteur, la réalisation d'un asservissement chimique comportant de la navigation autonome sur un robot hexapode de moins de 5 kg n'a pas été réalisé à ce jour. Nous verrons dans le chapitre suivant plus en détail la méthodologie employée pour répondre aux problématiques d'asservissement visuel

et chimique d'un robot hexapode. Ces algorithmes chimiques possèdent le point commun d'utiliser le flux d'air et la concentration du gaz pour converger vers sa source. La précision et leur efficacité en environnement réaliste, intérieur et avec des obstacles se dégradent, dû à ces perturbations (obstacles, flux d'air intermittent, etc.). Les environnements industriels, les carrières souterraines et les milieux contraints et dégradés sont sujet à ce type de configuration et engendrent des turbulences d'air ajoutant du bruit dans les mesures. Ce bruit impacte fortement la réalisation d'un asservissement chimique [208]. Nous avons vu ici plusieurs thématiques : l'architecture robotique, visuelle et chimique. Le modèle de robot sera un robot hexapode capable de reconnaissance automatique visuelle (méthode YOLO) et asservissement chimique (méthode par grille). Cependant, pour résoudre la problématique de reconnaissance d'anomalies dans les cas industriels, souterrains, navales et post-catastrophe ; un ensemble de fonctions est nécessaire : le contrôle de la marche, la localisation, la navigation, l'asservissement visuel et chimique. Nous présentons dans le prochain chapitre, la méthodologie adoptée pour ce travail de recherche.

MÉTHODOLOGIE

Dans ce chapitre, nous présentons une approche duale pour les percepts qui a pour objectif de détecter et se diriger vers la source d'un percept visuel et chimique dans un espace contraint et dégradé. L'approche de résolution du sujet est principalement issue des différentes cotations de critères des cas d'application (cas 1 : cuves de stockage et milieux navals, cas 2 : carrière souterraine, cas 3 : une zone urbaine post-catastrophe) dans le Chapitre I - Études des éléments contextuels (1.2). Nous avons évoqué précédemment, dans l'état de l'art, les différentes briques technologiques nécessaires à la réalisation de ce travail de recherche : un robot hexapode, un système de contrôle et de navigation, et enfin les méthodes d'asservissements visuels et chimiques. Dans un premier temps, nous présentons notre approche globale proposée. Par la suite, nous détaillons chaque brique constituant notre approche et nous exprimons les interactions entre ces briques.

3.1 Approche proposée

Dans cette section, nous évoquons les différents choix d'outils algorithmiques et mathématiques afin de répondre à la problématique. Nous décrivons les différents choix réalisés et une approche de travail pour la réalisation d'un asservissement visuel et chimique en utilisant un robot hexapode. Dans un premier temps, nous présentons les données nécessaires pour la méthode. Puis, nous formulons une approche globale composée de différentes briques. Ces briques sont définies et détaillées dans les sections qui suivent. Nous abordons leurs rôles dans la solution proposée.

3.1.1 Les problématiques de l'approche

Dans cette partie, nous allons nous intéresser à différentes parties de réalisation liées au sujet de reconnaissance automatique de points d'intérêt pour un robot d'inspection, hexapode, dans un environnement contraint et dégradé. En premier lieu, un robot hexa-

pode nécessite un contrôleur avant de s'avancer dans la navigation autonome. Le contrôle des pattes dépend de la géométrie du robot, des différents segments et des degrés de libertés minimum et maximum de chaque articulation. La géométrie du robot est la donnée d'entrée principale pour réaliser la cinématique directe (FK, F pour "Forward", K pour "Kinematic") et inverse (IK, I pour "Inverse", K pour "Kinematic"). Nous présentons ici les différentes problématiques de l'approche dans le tableau 3.1.1.

N°	Thème	Problématiques / Axes de progrès
1	Contrôle d'un robot hexapode	Déterminer les positions des fins d'effecteurs.
2	Contrôle d'un robot hexapode	Réaliser le contrôle des fins d'effecteurs pour la démarche.
3	Localisation d'un robot hexapode	Viabiliser l'odométrie du contrôleur pour les cas contraints.
4	Navigation d'un robot hexapode	La prise en compte de la géométrie du robot dans l'environnement navigable.
5	Asservissement d'un robot hexapode	Réaliser la détection et l'asservissement d'un objet visuel.
6	Asservissement d'un robot hexapode	L'évitement d'obstacle durant un asservissement chimique ou visuel.

Ces problématiques constituent le questionnement et les orientations technologiques et scientifiques du développement de la solution. Un tableau des problématiques et solutions possibles sont exposées, Figure 3.1.1.

Catégories	Problématiques	Solutions	Références
Contrôleur cinématique	Résolution de la position des fins d'effecteurs	OpenSHC	[211]
Odométrie	Calcul de l'odométrie dans un espace contraint	Laser Scan Matcher avec AMCL	[210]
Navigation	Chemin le plus court en espace contraint	Algorithme Dijkstra	[199]
Asservissement visuel	Détection automatisée d'objet	aRuCo et YOLO V5	[192] [193] [89]
Asservissement chimique	Recherche de source de gaz	Algorithme de grille "Grid"	[208]
Simulation chimique	Modélisation d'un panache de gaz avec application robotique	Gaden	[208]

3.1.2 Approche globale

Les données d'entrées communes à l'approche globale sont l'architecture du robot pour le contrôle, les données du capteur laser pour l'odométrie, la carte d'environnement pour la navigation. Les servomoteurs renvoient des informations sur l'état et les incréments appliqués au contrôleur moteur. Les données d'entrées spécifiques au procédé d'asservissement visuel sont : les données des capteurs visuels, un flux vidéo RGB et un flux vidéo de profondeur. L'asservissement chimique récupère les données des capteurs chimiques pour avoir la concentration du gaz et la direction du flux d'air. Les données de sortie souhaitées sont la position du robot par rapport à la carte, la position des percepts visuels et chimiques traduite en point de navigation et enfin, l'atteinte de ces percepts où la position du robot doit être le plus proche possible de la position des percepts. Notre choix s'est porté sur l'utilisation d'un robot hexapode marchant sur 3 degrés de liberté par patte pour résoudre notre problématique détection et reconnaissance des POI.

L'hypothèse proposée se focalise sur le traitement du flux des capteurs (visuels et chimiques) dans l'objectif d'établir une loi de commande permettant de ressortir un vecteur d'approche. Ce vecteur d'approche a pour rôle fondamental de se rapprocher du percept. Ce vecteur est transformé en point de navigation. Le robot va alors s'approcher de manière continue (asservissement visuel) et de manière itérative (asservissement chimique) vers les percepts à condition qu'il soit détectable / détecté et les informations suffisamment consistantes. Nous proposons deux méthodes de traitement séparé pour une approche d'asservissement visuel et chimique. Ce choix est justifié sur le fait que le percept visuel est statique sur l'environnement (des défauts sur une surface, infrastructure, etc.), tandis que le percept chimique, est de par sa nature, dynamique et évolue au sein de l'environnement. L'apport de cette thèse est de lier la navigation aux processus d'asservissements et de proposer deux méthodes permettant le traitement des différents percepts. Nous proposons 6 différentes briques : le contrôleur du robot, l'odométrie, la navigation, l'asservissement chimique, la détection d'objets automatisée et l'asservissement visuel.

Le contrôleur du robot (hexapode) a pour rôle de réaliser les calculs cinématiques pour le contrôle des pattes et assurer le déplacement du robot d'un point vue mécanique. L'odométrie va permettre de calculer le déplacement effectué par le robot et de pouvoir localiser le robot par rapport aux mouvements qu'il aura effectués dans l'environnement. La navigation utilise les données odométriques, les vecteurs de déplacement du contrôleur et la carte pour permettre au robot de se déplacer par le biais de points de navigation.

Les asservissements (visuels et chimiques) utilisent les données des capteurs pour fournir un point de navigation. Enfin, la détection visuelle est la méthode employée pour détecter les points d'intérêts visuels. Le schéma de l'approche globale proposée est donné par la Figure 3.1).

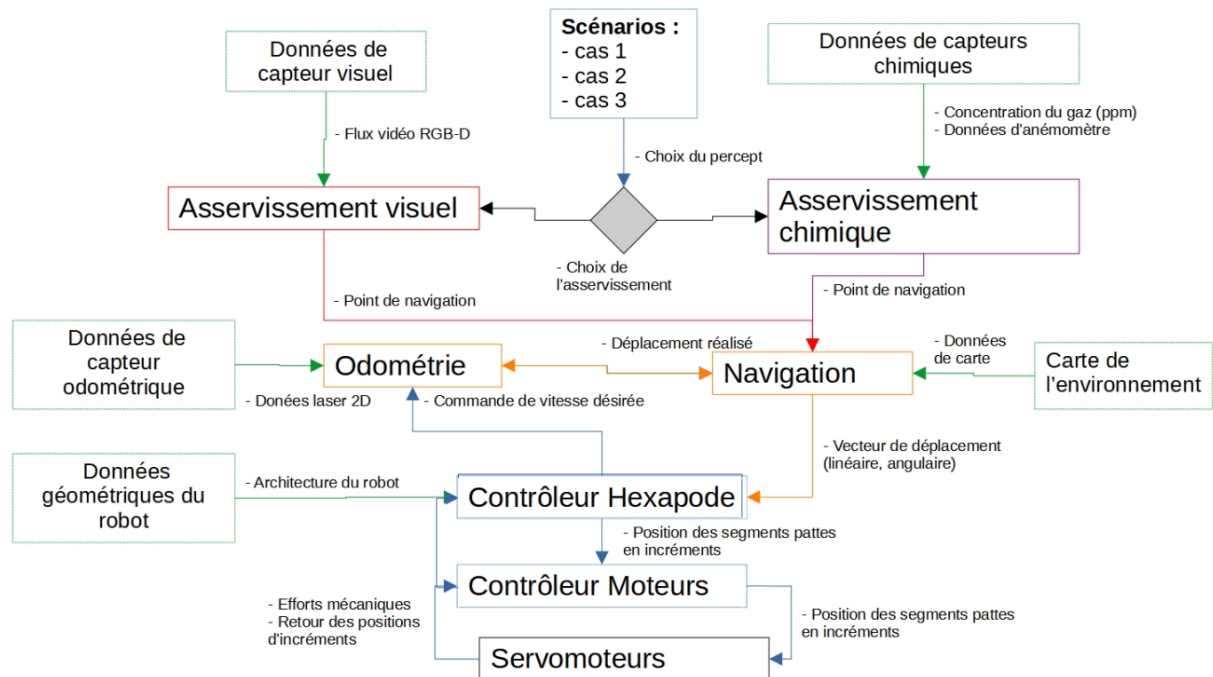


FIGURE 3.1 – Approche proposée pour les données capteurs (en vert), le contrôle (en bleu), la navigation (en orange) et l'asservissement visuel (en rose) et chimique du robot hexapode (en violet).

Nous abordons par la suite le détail des briques suivantes : le système de contrôle pour le robot hexapode, la navigation, l'asservissement visuel et chimique.

3.2 Brique de contrôle de l'hexapode

Dans la partie contrôle, nous abordons les méthodes permettant de contrôler la pose et la démarche du robot par rapport à sa morphologie (dimensions des segments, angles minimaux et maximaux, etc.). Le contrôleur est la pierre angulaire réalisant le lien entre la navigation et les effecteurs (servomoteurs) comme nous le montre la Figure 3.2. Afin de pouvoir contrôler le robot hexapode, la résolution du positionnement des fins d'effecteurs doit être effectuée en fonction des angles de rotation pour chaque degré de liberté. Pour cela, la cinématique directe et inverse est employée et décrite dans ce chapitre.

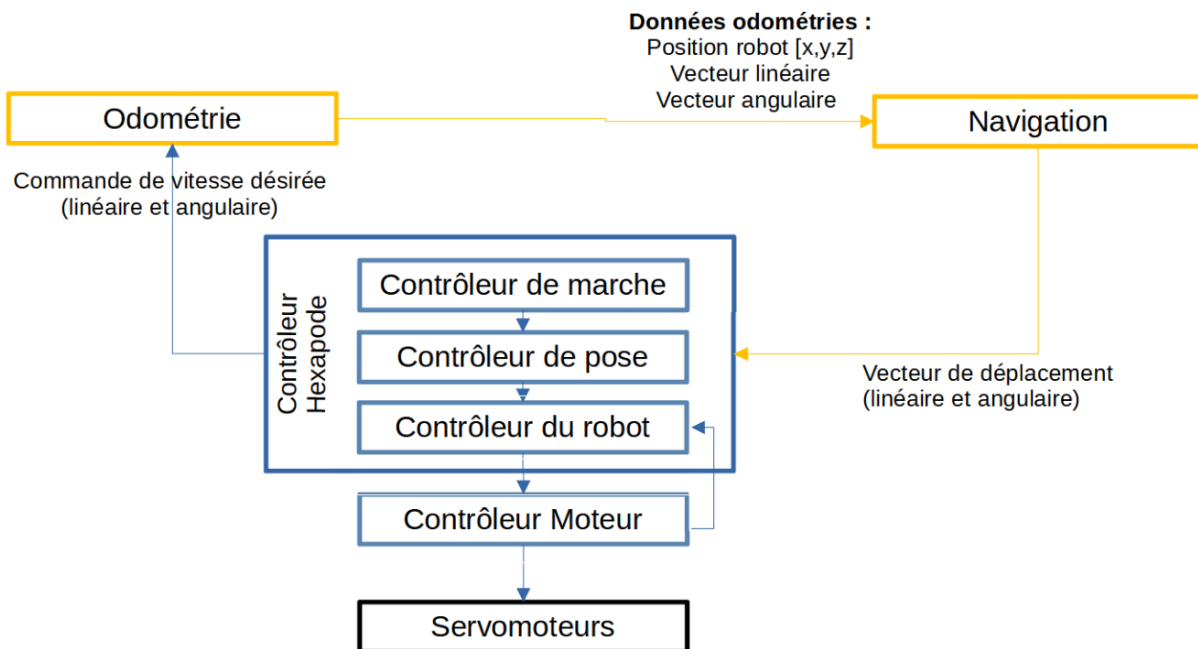


FIGURE 3.2 – Approche proposée pour le contrôle du robot hexapode. En bleu : la brique de contrôle, en jaune : la brique de navigation et en noir : les servomoteurs.

Sur le schéma 3.2, la brique de contrôle de l'hexapode est composée de 3 éléments. Le contrôleur de pose va permettre d'ajuster les positions des pattes à leurs extrémités (fin d'effecteurs) en fonction des différentes poses statiques (repos, debout, etc.). Les poses dynamiques sont alimentées par le contrôleur de marche. Les démarches sont générées sous forme de séquences permettant au robot de se déplacer par une succession de différentes poses et donc une succession des différents points d'application des fins d'effecteurs. La communication de la démarche se traduit d'une séquence de positions de fins d'effecteurs à une séquence d'angles en radians. Le contrôleur du robot permet de traduire ses angles en

radians en position d'incrément pour le contrôleur des moteurs. Cette opération dépend de la géométrie du robot, de la nature des servomoteurs et des paramètres du robot en général. Le contrôleur du moteur permet, quant à lui, de faire l'interface entre le matériel (les servomoteurs) et le contrôleur de l'hexapode (informations de contrôle d'angles). Il sert d'interface pour la communication, fait office de pilote et permet de faire remonter les informations motrices au contrôleur de l'hexapode. Le contrôleur moteur dépend du fabricant des servomoteurs et de la suite logiciel qu'il met à disposition.

3.2.1 Système de contrôle d'un robot hexapode

Dans cette section, nous allons évoquer l'aspect contrôle du robot hexapode en présentant l'approche détaillée de l'architecture, les méthodes de résolution cinématique et enfin la démarche. Comme évoqué précédemment, le contrôleur est utilisé pour générer des positions de fins d'effecteur (contrôleur de poses), pour générer la démarche du robot (contrôleur de marche) et le plus important, pour générer le modèle cinématique (contrôleur du robot). L'architecture choisie est un robot hexapode (3 paires de pattes), figure 4.3, avec 3 degrés de liberté sur chaque patte. La locomotion des robots hexapodes nécessite une séquence de mouvements spécifique pour aller tout droit, reculer, tourner en sens horaire, antihoraire, mais aussi translater à gauche et à droite (mouvement holonome).

Dans notre cas, le robot hexapode aura 3 DoF par patte, soit 18 actionneurs à synchroniser pour produire le mouvement voulu. La coordination des pattes est un mécanisme permettant le contrôle de la transition entre les séquences de mouvement de pattes et produisant la démarche. Pour réaliser la démarche, la première étape est de considérer le modèle géométrique du robot. Notamment les longueurs des différents segments (coxis, fémur, tibia). Le modèle de la patte et les différentes longueurs de segment ont une influence sur les efforts exercés sur les coxis, fémurs et tibias. Le couple des différents degrés de liberté sera modélisé sur le robot réel à travers un jumeau numérique (la représentation du modèle mathématique et géométrique). Afin de réaliser le modèle mathématique et géométrique du robot, il convient de modéliser la cinématique du robot.

Cinématique d'un robot hexapode

L'architecture d'une patte (figure 3.3) est définie par ces 3 segments : le coxis, l'articulation reliée au corps permettant les translations et les rotations du corps, le fémur

et le tibia servent à réaliser les transitions entre les appuis et les foulées. Les angles des différents segments : coxis, fémur, tibia, sont respectivement et usuellement appelés θ_1 , θ_2 et θ_3 . La cinématique inverse d'un robot hexapode est définie par la position théorique des fins d'effecteurs souhaitée dans le but d'ajuster les différents angles θ des segments de manière automatique.

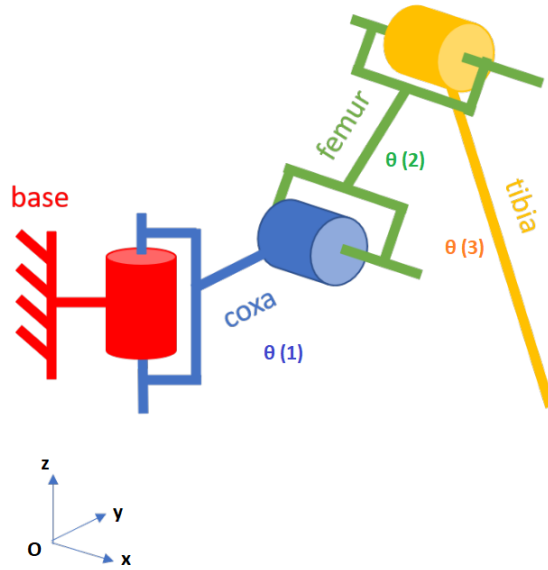


FIGURE 3.3 – Concept de modélisation d'une patte.

La méthode Denavit-Hartenberg

La méthode Denavit-Hartenberg permet de calculer la position relative entre deux solides à partir de 4 paramètres (θ, α, d, i) . Le modèle du robot peut être représenté en utilisant la matrice de Denavit-Hartenberg qui permet de réaliser facilement les transformations liées à la cinématique directe et inverse sur les angles des différents joints et des fins d'effecteurs. Le corps du robot peut être représenté par sa pose (o_b, x_b, y_b, z_b) , son orientation $(o_{roll}, o_{pitch}, o_{yaw})$ et sa vitesse $(\dot{x}, \dot{y}, \dot{z})$. Le modèle du robot fournit aussi l'odométrie du robot par rapport à la référence de la cartographie de l'environnement (o_m, x_m, y_m, z_m) qui est considéré comme fixe. Chaque patte possède ses propres coordonnées locales (o_p, x_p, y_p, z_p) . Les quatre paramètres a_i , α_i , d_i et θ_i reçoivent généralement et respectivement les noms de longueur de lien, de torsion de lien, de décalage de lien et d'angle de joint.

Afin de décrire quantitativement le robot avec un modèle mathématique, une série de systèmes de coordonnées doit être établie. Un système de coordonnées universel est établi pour décrire la locomotion du robot, désigné par le symbole $\Sigma Ow(Xw, Yw, Zw)$. Ensuite, les systèmes de coordonnées du corps et de référence des liens de jambe sont établis, désignés par les symboles $\Sigma Ob(Xb, Yb, Zb)$ et $\Sigma O_0^i(X_0^i, Y_0^i, Z_0^i)$ ($i=1,2,3,4,5,6$), respectivement. Des systèmes de coordonnées attachés à l'articulation du coxis (β_1^i), à l'articulation du fémur (β_2^i) et à l'articulation du tibia (β_3^i) de la jambe i sont établis pour chaque articulation. Le modèle hexapode simplifié et les systèmes de coordonnées sont illustrés figure 3.4.

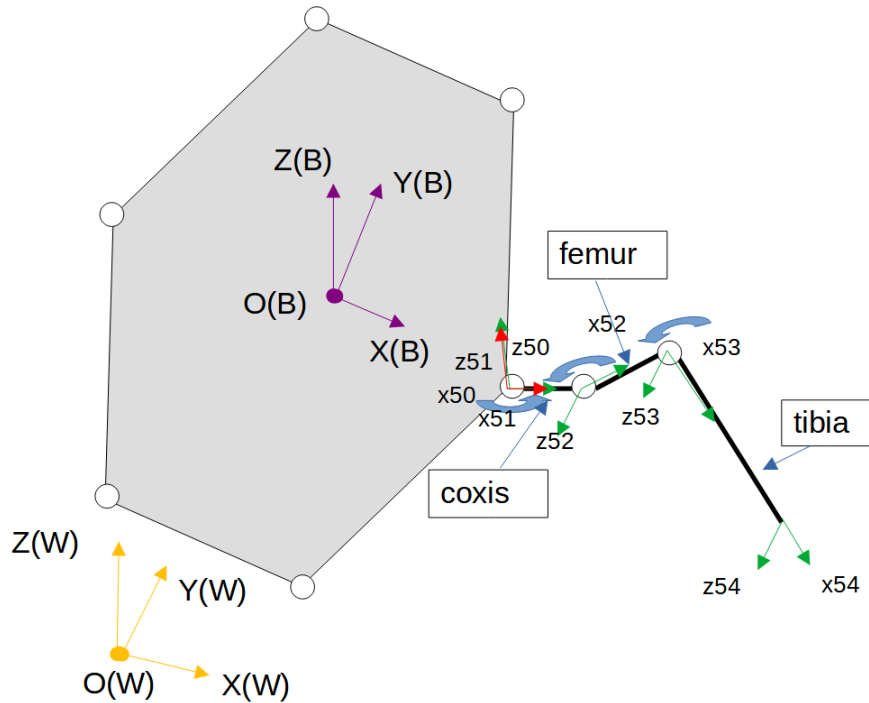


FIGURE 3.4 – Modélisation d'une patte d'hexapode avec les systèmes de coordonnées.

La matrice Denavit-Hartenberg théorique est exposée ici 3.1 :

$$\begin{bmatrix} \text{Joint}, & a, & \alpha(^{\circ}), & d, & \beta(^{\circ}), \\ 1 & 0, & 0, & 0, & \beta_1 \\ 2 & l1, & 90, & 0, & \beta_2 \\ 3 & l2, & 0, & 0, & \beta_3 \\ 4 & l3, & 0, & 0, & 0 \end{bmatrix} \quad (3.1)$$

La cinématique directe

La problématique de la cinématique directe consiste à exprimer les coordonnées spatiales en fonction des variables articulaires d'un robot. En l'occurrence, les variables articulaires sont composées des différents angles θ_1 , θ_2 et θ_3 . On emploie généralement la méthode DH pour dériver un ensemble d'équations cinématiques qui donne la position et l'orientation de l'effecteur d'extrémité du robot suivant les paramètres d'articulation. Les équations dérivées ont été utilisées par le planificateur de pas (contrôleur de pose) pour calculer l'espace de travail de la jambe du robot et pour projeter des configurations finales pour les fins d'effecteurs pour engager la démarche (contrôleur de marche). La convention Denavit-Hartenberg (DH) stipule que chaque transformation homogène, H_i , peut être exprimée comme un produit de 4 transformations concernant les 4 paramètres DH (θ, α, r, d), donné par l'équation 3.2 :

$$H_i = R(Z_i, \theta_i), T(Z_i, d_i)T(X_i, a_i)R(X_i, \alpha_i) \quad (3.2)$$

Où R et T désignent respectivement la rotation et la translation. L'équation (3.2) montre la matrice qui a été calculée à partir du produit de 4 matrices de transformation données dans (3.2). La transformation de matrice pour les paramètres Denavit-Hartenberg θ_i est donnée par :

$$Rot_{z, \theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

La transformation de matrice pour les paramètres Denavit-Hartenberg d_i est donnée par :

$$Trans_{z, d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

La transformation de matrice pour les paramètres Denavit-Hartenberg a_i est donnée

par :

$$Trans_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

La transformation de matrice pour les paramètres Denavit-Hartenberg α_i est donnée par :

$$Rot_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\theta_i & -\sin\theta_i & 0 \\ 0 & \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$H_i = R(Z_i, \theta_i), T(Z_i, d_i)T(X_i, a_i)R(X_i, \alpha_i) \quad (3.7)$$

La cinématique directe de l'hexapode nous permet de déplacer les segments en fonction d'un angle donnée pour θ_1 , θ_2 et θ_3 .

La cinématique inverse

Les calculs de la cinématique inverse sont présentés sous les équations [211]. La cinématique inverse calcule le changement souhaité de l'angle d'articulation ($\Delta\theta$) (dans la notation 3.2.1) pour un changement incrémentiel donné de la position de l'effecteur final ($\Delta\vec{s}$) (dans la notation 3.2.1) est calculé en utilisant la matrice Jacobienne J et la méthode de Levenberg-Marquardt, également connue sous le nom de méthode des moindres carrés amortis est utilisé pour optimiser la fonction de coût [133]. La dérivation complète est fournie par les équations (3.8), (3.9) et l'équation résultante à résoudre donnée par les équations (3.17) page 105 à 106.

Le déplacement souhaité dans les angles articulaires s'écrit : $\Delta\theta$. Le changement incrémentiel de la position de l'effecteur final correspond à : $\Delta\vec{s}$. Le changement des angles articulaires est calculé en utilisant le Jacobien pour la position de l'effecteur final. La matrice Jacobienne J (3.8) est définie comme :

$$J(\theta) = \left(\frac{\delta s_i}{\theta_j} \right)_{i,j} \quad (3.8)$$

Où s_i est l'effecteur final du lien, en utilisant les joints θ_j comme paramètres. Ensuite, le vecteur de rotation instantanée du Jacobien (3.9) peut être calculée :

$$\frac{\delta s_i}{\delta \theta_j} = v_j \times (s_i - p_j) \quad (3.9)$$

Où v_j est le vecteur unitaire le long de l'axe de rotation courant et p_j est la position du j^{th} joint. Pour les itérations incrémentielles de l'effecteur final, un changement dans l'effecteur final peut être approximé par :

$$\Delta \vec{s} \approx J \Delta \theta \quad (3.10)$$

L'utilisation de la méthode Levenberg-Marquardt (méthode des moindres carrées) permet de résoudre $\Delta \theta$ en s'approchant du conditionnement de la matrice (robustesse face au bruit) en utilisant un paramètre d'amortissement. λ correspond au facteur d'amortissement positif qui est ajusté à chaque itération. Si l'itération fait diminuer s rapidement, on peut utiliser une valeur plus faible pour λ . Ce qui oriente l'algorithme dans une optique Gauss-Newton. Or, si une itération est peu concluante, l'augmentation de λ oriente l'algorithme dans une optique similaire à celui du gradient. Cette utilisation permet de résoudre certaines problématiques de régression linéaires et non-linéaire et trouve une solution même dans le cas où l'algorithme est initialisé loin d'un minimum. L'expression est décrite ci-dessous :

$$\Delta \theta \approx J^T (J J^T + \lambda^2 I)^{-1} \Delta \vec{s} \quad (3.11)$$

Où $\lambda > 0 \in \mathbb{R}$.

Pour optimiser la cinématique du robot, nous devons prendre en considération les stratégies d'évitement de la limite de position/vitesse (JLA) de l'articulation pour augmenter la sécurité de l'articulation pendant les opérations de marche. La fonction de coût peut être décrite ici (3.12), afin de minimiser la norme d'un vecteur (p-norme), et d'approcher la focalisation sur le joint, le plus éloigné de son centre :

$$\Phi(q) = \left(\sum_{i=1}^n |K_{ii} \frac{q_i - q_{ci}}{\Delta q_i}|^p \right)^{\frac{1}{p}} \quad (3.12)$$

Où q_{ci} est le centre de la plage commune des joints Δq_i avec comme joint i , et K est la matrice des poids des joints. q_i représente les différents joints de chaque patte.

La solution optimisée est décrite (3.13) :

$$v = -\nabla\Phi \quad (3.13)$$

v représente la vitesse des joints. L'équation (3.13) peut être optimisée pour une fonction combinée (3.14) :

$$q\theta = Z\Delta\vec{s} + (I - ZJ)v \quad (3.14)$$

l_s (longueur de la foulée) et f_s (la fréquence des pas) ne sont pas totalement indépendants, f_s est limité par la vitesse articulaire, également impactée par la distance nécessaire pour s'éloigner de la longueur de foulée. Le vecteur est le long de l'axe de rotation courant et p_j est la position du $j^{\text{ème}}$ joint. La vitesse du corps maximal (3.15) est calculé avec le facteur de service (duty factor) :

$$v_{body} = \frac{l_s \times f_s}{\beta} \quad (3.15)$$

v_{body} représente la vitesse du corps de l'hexapode. Où β , le facteur de service ("duty factor") est définis par le rapport entre T_{stance} (phase d'appui) et T_{stride} (phase de marche / foulée) :

$$\beta = T_{stance}/T_{stride} \quad (3.16)$$

, le temps dans la phase d'appui et le temps total de la phase d'appui et de balancement pour le temps de foulée.

Où $\lambda > 0 \in \mathbb{R}$, nous utilisons un compact et une approximation, le résultat final est affiché sur (3.17) :

$$\Delta\theta = J^T(JJ^T + \lambda^2 I)^{-1}\Delta\vec{s} = Z\Delta\vec{s} \quad (3.17)$$

Où $Z = J^T(JJ^T + \lambda^2 I)^{-1}$

Les résolutions de la cinématique directe et inverse permettent de relier les angles des différents segments de pattes, et ainsi relier la base du robot comme référence (base mobile). Une fois l'acquisition des coordonnées de fins d'effecteurs, il faut générer l'espace de travail des différentes pattes et déployer une stratégie d'ordonnancement pour permettre le déplacement : la démarche.

Contrôle dynamique d'un robot hexapode

La démarche est le procédé permettant le déplacement du corps par la succession de nouveaux points d'appuis. Le déplacement des points d'appuis dans une direction entraîne le déplacement du corps dans cette même direction. La morphologie du marcheur et la manière d'effectuer la démarche ont un impact direct sur la vitesse et la stabilité. Deux types de démarches existent présentant chacune des avantages et inconvénients : la marche quasi-statique et la marche dynamique. La marche dynamique se caractérise par des effets d'accélération et d'inertie très marqués. Les perturbations d'équilibre sont utilisées pour produire des déplacements. Le robot hexapode a donc son centre de masse en dehors de sa plage de stabilité et se sert de la quantité de mouvement déjà acquise pour poursuivre son mouvement. Cette démarche est ainsi une succession de déstabilisations contrôlées pour produire l'élan nécessaire pour avancer.

L'oscillation de la marche, similaire à l'effet de pendule, convertit l'énergie cinétique de la masse en énergie potentielle et converti celle-ci en énergie cinétique par la suite. La marche dynamique possède quelques inconvénients. Des corrections, liées au contrôle, doivent être réalisées en permanence, afin d'assurer un mouvement fluide. La résolution des différentes équations cinématiques demande une certaine puissance de calcul à cause des différents paramètres à prendre en compte. De plus, cette démarche est adaptée sur des grandes distances sans arrêt fréquents. Étant donné la problématique et le procédé d'inspection, cela rend ce type de marche inadapté.

Au contraire, la marche quasi-statique correspond aux cas où les effets d'accélération et d'inerties soient négligeables. Cette démarche permet d'obtenir une succession de positions stables à tout moment et de ce fait, permet d'arrêter le robot en pleine démarche en restant stable. La cinématique et la dynamique permettent d'analyser la mécanique de la marche. La cinématique vise à définir la configuration des différents membres afin de calculer la position des fins d'effecteurs. L'expression des efforts est une thématique importante des robots marcheurs. Elle peut être décrite avec plusieurs valeurs telles que la force, la masse et l'accélération. Comme décrit dans l'équation 3.18

$$\Sigma F = \Sigma m \cdot a \quad (3.18)$$

$$\Sigma C = \Sigma (I\alpha + I\omega) \quad (3.19)$$

F est la force, m la masse, a l'accélération, C le couple, I l'inertie, α l'accélération

angulaire et ω est la vitesse angulaire. Le positionnement d'un robot marcheur statique dans l'espace (position et orientation) dépend des termes de l'accélération, qui doivent être alors nuls, et les sommes des forces et des moments par rapport à chacun des axes doivent être égales à zéro. Un corps dans l'espace possède six degrés de liberté (translation sur x , y , z et rotation sur x , y , z), nous avons donc 6 équations scalaires.

La démarche d'un robot hexapode

Comme nous l'avons vu précédemment, un robot hexapode possède différents régimes de démarche, de manière à osciller les pattes pour produire un déplacement. La démarche se compose en trois étapes (Figure 3.5 [211]). La démarche est défini suivant 3 couleurs : le jaune pour la position de démarrage, le bleu pour la position de repos et le vert pour la phase de transition / balancement. La position de démarrage (en jaune), toutes les pattes sont au sol et le passage au stade d'oscillation (mouvement noté en vert) indique les pattes en mouvement, quittant le sol. Enfin, les pattes qui ne bougent pas et sont au sol, sont indiquées en bleu. Les robots marchants possèdent plusieurs allures et démarches différentes. L'hexapode possède 5 démarches principales différentes, mais d'autres séquences peuvent être générées artificiellement ou manuellement.

La démarche par vagues (Figure 3.5) est une démarche par oscillation unitaire. Chaque patte va se déplacer une à une. La démarche est lente comparée aux autres, mais elle permet une meilleure stabilité sur les terrains difficiles. La démarche souple ("amble gait") sollicite deux pattes opposées géométriquement à chaque fois et en même temps sur un rythme régulier et sans temps mort. C'est une démarche équilibrée entre stabilité et vitesse qui permet les grandes foulées.

La démarche ondulante ("ripple gait") est une démarche semblable à la démarche souple, à l'exception qu'une alternance de mouvements est réalisée. Une oscillation d'une patte est réalisée juste avant de poser le couple de pattes au sol. C'est un mouvement très arachnéen d'un point de vue cinématique. C'est une démarche dynamique qui permet d'adapter des foulées lentes comme des foulées plus rapides. C'est une démarche habile pour le franchissement d'obstacles, mais le mouvement de la patte seule peut potentiellement renverser l'hexapode si elle s'appuie sur la surface verticale de l'obstacle.

La démarche tripode ("tripod gait") met en mouvement 3 pattes à chaque fois et sans chevauchement de période d'oscillation. C'est une démarche dynamique, mais potentielle-

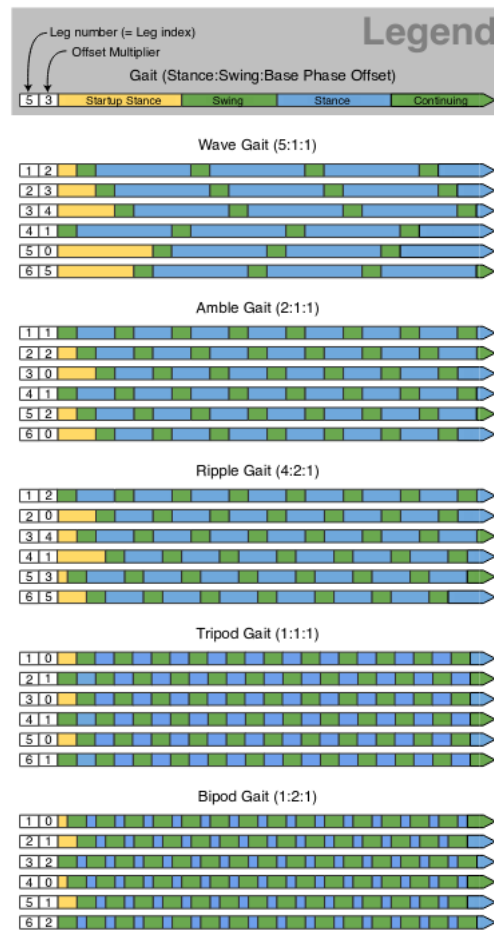


FIGURE 3.5 – Démarche d'un robot hexapode des travaux de Tam et al. [211].

ment critique en stabilité et donc déconseillée pour du franchissement. En effet, lors d'une démarche où 3 pattes sont en mouvement, le centre de gravité peut être à l'extrémum du polygone de support. Pour un hexapode mal équilibré, le centre de gravité est de ce fait excentré et dégrade la stabilité.

La démarche du bipode ("bipod gait") sollicite un mouvement fluide alterné d'une paire de pattes avec un chevauchement d'oscillation. C'est une démarche inspirée de la locomotion des bipèdes. Pour un robot avec 6 pattes au lieu de 2, c'est une démarche plus rapide que la démarche à vague.

Forts de ce constat, nous utiliserons la démarche ondulante pour les phases de locomotion longue et la démarche par vagues sera utilisée pour le franchissement d'obstacle. Dans cette partie, nous avons décrit la résolution de la cinématique d'un robot hexapode

permettant d'aborder les notions de démarches. Le contrôle des fins d'effecteurs est des enjeux majeurs des robots marchants. Cependant, le simple contrôle d'un robot hexapode ne suffit pas pour assurer la navigation dans un milieu contraint et dégradé.

3.3 Brique de navigation d'un robot hexapode

La problématique de la navigation d'un robot dans un milieu confiné est un thème répandu dans le domaine de la robotique mobile. La problématique de navigation en espace contraint pour un robot hexapode pose la question de la gestion de l'évitement d'obstacle. Cette partie est composée de plusieurs éléments : la trajectoire du robot, l'empreinte du robot dans son espace et les fonctions de coût associé à la carte. L'aspect contrôle et démarche d'un hexapode constitue la première étape vers l'autonomie. Le robot pouvant maintenant se déplacer, il faut mesurer avec précision ses déplacements afin de pouvoir le positionner sur la carte. Une source odométrique provenant de la cinématique est une donnée approximée de la réalité et peut cumuler de l'erreur en fonction du sol, de la pente, ou d'autres aspérités au niveau du sol.

La navigation sert à apporter au robot l'autonomie nécessaire pour lui permettre de se déplacer de manière automatisée dans un environnement grâce à des points de navigation placés dans un repère "carte". Le rôle de la navigation dans notre approche va permettre au robot de pouvoir 1) se positionner par rapport à l'environnement (les obstacles, les lieux navigables), 2) traiter les différentes informations de la carte et 3) positionner les points d'intérêts sur la carte. Dans la partie navigation, nous abordons l'approche de l'odométrie, les liens avec le contrôleur de l'hexapode et le traitement des points de navigation en commande de vitesse. Afin de réaliser ce procédé, les données d'entrées nécessaires sont l'algorithme de localisation (AMCL), l'odométrie, les transformées des capteurs (repère cartésien), les données de capteur laser 2D et la carte. Les fonctions de coût local et global sont utilisées pour calculer les espaces occupés, libre et espace inconnu.

Le planificateur global utilise le coût global pour générer un plan sur le long terme. Alors que le planificateur local utilise le coût local pour générer un plan sur le court terme. Les comportements de récupération sont des actions réalisées lorsque le robot est perdu dans la carte ou bloqué (par exemple, tourner autour de lui-même). L'algorithme de trajectoire permet de calculer le chemin le plus court entre le robot et le point de navigation souhaité. Nous présentons ici l'approche de la navigation Figure 3.6. Cette approche est composée en 4 parties : l'algorithme de trajectoire, la planification locale et globale (paramètres du robot), la fonction de coût local et global (lié à l'encombrement du robot).

La problématique de localiser un robot dans un environnement contraint et dégradé est un domaine actif dans la recherche robotique. Comme nous l'avons vu dans le Chapitre

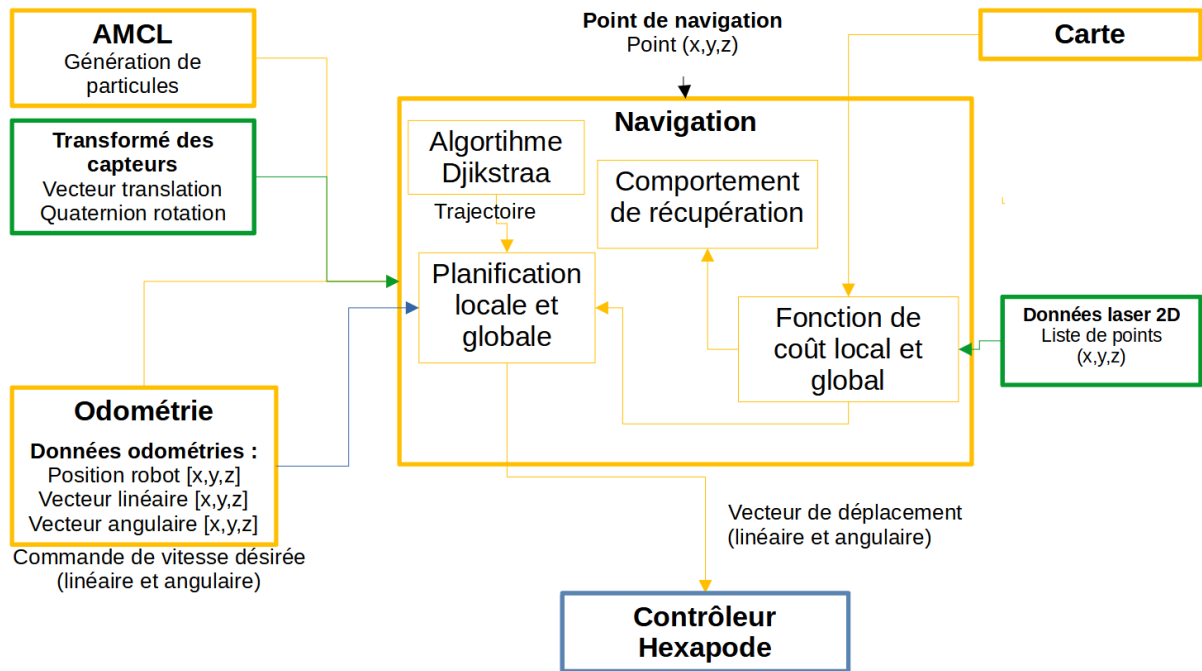


FIGURE 3.6 – Approche proposée pour la navigation du robot hexapode. En jaune : la brique de navigation, en bleu : le contrôleur de l'hexapode et en vert : les données capteur

I : Analyse Contextuelle, l'environnement peut influencer sur les capteurs et ajouter du bruit (par exemple : les parois métalliques d'un ouvrage industriel (cas 1), dans une carrière souterraine ferromagnétique (cas 2), etc.).

Nous avons évoqué précédemment la problématique de l'odométrie pour un robot marcheur, hexapode. En effet, le calcul de l'odométrie par le contrôleur n'est pas toujours fiable et dépend de la vitesse d'oscillation des pattes, de la démarche et des bruits résiduels mécaniques (terrain accidenté, franchissement d'obstacle, etc.). L'odométrie joue un rôle capital dans la navigation d'un robot, elle permet d'estimer la trajectoire de déplacement effectuée et la position du robot dans le référentiel de la carte ("monde").

3.3.1 L'odométrie

La problématique de l'odométrie calculée par le contrôleur d'un robot marcheur est qu'elle est imprécise dans le cas d'une boucle de contrôle ouverte. En effet, ce calcul peut être généralisé sur la démarche pour deux conditions. La première : le corps est supporté par au moins 3 pattes avec des pieds coplanaires et les pattes ne subissent pas de glis-

sement [62]. Dans ce cas, la vitesse corporelle souhaitée est transmise au robot pour une fréquence de pas donnée, avec la longueur de foulée correspondante pour calculer et atteindre la vitesse du corps souhaitée [211]. Le maximum de la vitesse linéaire du corps est donc limité par l'espace de marche des jambes. Les sources odométriques peuvent être de 2 sources différentes ou complémentaires : les capteurs proprioceptifs, mesurant l'accélération linéaire et angulaire du robot (centrale inertielle), et les capteurs extéroceptifs, mesurant le robot à partir de son environnement. L'odométrie sert à calculer la position du robot dans un espace en fonction de son architecture, ses capteurs (proprioceptifs ou extéroceptifs). Un robot de type holonome se comporte différemment d'un robot non-holonome (à roues ou à chenille par exemple) au niveau de ses possibilités de déplacement (déplacement sur l'axe y) ainsi que sur son mode (réaliser l'odométrie sur une démarche). Le calcul odométrique peut être généré par le biais du contrôleur du robot.

Odométrie par laser

La robustesse de la position d'un robot hexapode est donc primordiale dans les différents cas à traiter. L'odométrie nécessite d'être fiable pour mettre à jour la localisation du robot en évolution dans son environnement. Lorsque cette odométrie est imprécise, le robot dérive sur sa localisation à cause d'une erreur cumulée ("Drifting effect") et/ou est déplacé à un endroit arbitraire de la carte ("Kidnapping effect"). L'approche utilisée dans ce travail de recherche emploie un capteur laser 2D (RPLidar) pour faire une correspondance entre deux scans lasers pour calculer la localisation du robot dans son espace [210]. Cette méthode a été utilisée pour améliorer l'algorithme de localisation Monte-Carlo [194], évoqué au Chapitre 2. La problématique de cet algorithme est que la résolution d'un changement de pose discontinue n'est pas possible. La localisation va alors échouer [196] et le robot sera déplacé aléatoirement dans son espace, le problème du "kidnapping" [194].

Le modèle proposé est basé sur des commandes envoyées au robot pour prédire sa position. Les commandes du robot sont composées de translation et de rotation (Paramétrées par les angles d'Euler (x,y,z) par rapport au robot) exprimées dans le cadre local du robot.

Par conséquent, le vecteur d'entrée prend la forme de :

$$u(t) = [t_x \ t_y \ t_z \ \alpha \ \beta \ \gamma] \times T \quad (3.20)$$

Où t_x, t_y, t_z sont les translations attendues et $\alpha\beta\gamma$ sont les différentes rotations (r)

suivant les axes x , y et z . Le mouvement du robot est modélisé par la fonction de prédiction suivante :

$$x_r^{new} = \begin{bmatrix} r + R_W^R t \\ q \times q_\alpha \times q_\beta \times q_\gamma \end{bmatrix} \quad (3.21)$$

Où $R_W^R t$ est la matrice de rotation entre le cadre global (la carte de l'environnement) et le cadre du robot, t est le vecteur de translation, q_α , q_β , q_γ représentent les rotations des angles en quaternion et \times , le produit d'Hamilton. Comme il n'y a pas de boucle fermée dans la commande de mouvement, le robot produit un déplacement différent de celui attendu. Cette incertitude d'entrée (causée principalement par le glissement de la jambe du robot et l'imprécision des servomoteurs) est modélisée comme une fonction gaussienne multivariée et est représentée par une matrice de covariance P_i .

Actuellement, on suppose que les entrées sont indépendantes et que P_i est une matrice diagonale. Les Jacobiens de la fonction de prédiction sur l'état du robot et les entrées du robot sont utilisées pour augmenter la robustesse du système. La localisation du système peut ainsi être plus robuste. Ce procédé sera utilisé dans la partie navigation et permettra de garder la localisation fiable quelle que soit la trajectoire générée par le robot dans un espace contraint et dégradé. L'odométrie nous fournit une localisation, qui peut être imprécise en fonction de la situation de l'environnement. Afin d'améliorer la robustesse de cette donnée, nous proposons l'ajout d'une méthode probabiliste.

3.3.2 AMCL : Localisation par adaptation Monte Carlo

La problématique de localisation des environnements contraints pour un robot hexapode peut être résolu par une approche probabiliste. Une approche adaptée pour les environnements contraints a été proposée [194] en utilisant AMCL (Adaptative Monte Carlo Localization) et un capteur LIDAR. AMCL est une méthode de localisation probabiliste pour un robot se déplaçant en 2D. Elle met en œuvre l'approche de localisation Monte Carlo adaptative (ou KLD-sampling) [195], qui utilise un filtre à particules pour suivre la pose d'un robot par rapport à une carte connue. L'algorithme de localisation de Monte Carlo est fondamentalement divisé en quatre étapes :

- **Étape 1** : Initialiser l'essaim de particule.
- **Étape 2** : Simuler les particules en mouvement.
- **Étape 3** : Calculer le score des particules.
- **Étape 4** : Échantillonnage des particules.

Lorsqu'un capteur laser est utilisé pour localiser le robot sur une carte 2D, si la pose du robot est donnée, il est très facile de calculer la correspondance entre les faisceaux laser et la distance séparant le robot des contours de la carte. Par conséquent, l'algorithme AMCL peut être utilisé, ce qui représente la pose du robot avec de nombreuses particules. Cela permet de calculer le poids de la particule en fonction de l'occupation de la carte pour déterminer la pose estimée et localiser le robot. Cependant, il y a quelques problèmes dans l'algorithme AMCL, il ne peut pas résoudre le problème d'enlèvement de robot. Une fois que la pose change, la localisation échouera. Pour améliorer la précision de la localisation, de nombreuses particules doivent être ajoutées et une fusion des capteurs (IMU et Laser 2D) pour ajouter davantage de précision.

L'algorithme AMCL [194] est adapté de l'algorithme MCL pour résoudre les problèmes ci-dessus. Il ajoute aléatoirement des particules libres lors du ré-échantillonnage. Le nombre de particules libres est calculé sur la base de poids estimés ω_{lent} estimé lent et rapide ω_{rapide} :

$$\omega_{lent} = \omega_{lent} + \alpha_{lent}(\omega_{moy} - \omega_{lent}) \quad (3.22)$$

$$\omega_{rapide} = \omega_{rapide} + \alpha_{rapide}(\omega_{moy} - \omega_{rapide}) \quad (3.23)$$

Dans les équations 3.22 3.23, ω_{moy} est le poids moyen de toutes les particules ; les paramètres α_{lent} et α_{rapide} sont respectivement utilisés pour estimer le taux d'atténuation du filtre à long terme et le poids à court terme. L'évolution lente et rapide sont respectivement le taux de décroissance exponentielle pour le filtre de poids moyen lent et rapide, utilisé pour décider quand récupérer en ajoutant des poses aléatoires.

3.3.3 Gestion de la planification de trajectoire

La gestion de la trajectoire revient, a priori, à résoudre le problème du chemin le plus court ou encore appelé problème du "voyageur de commerce". Ce problème se décline en plusieurs variantes. La plus simple est la suivante : étant donné un graphe non orienté, dont les arêtes sont munies de poids et deux sommets de ce graphe, trouver un chemin entre les deux sommets dans le graphe, de poids minimum. L'algorithme de Dijkstra [199] permet de résoudre un problème plus général : le graphe peut être orienté, et l'on peut désigner un unique sommet, et demander d'avoir la liste des plus courts chemins pour tous les autres nœuds du graphe.

Les paramètres importants à prendre en compte sont la facilité d'utilisation, les performances odométrique et le coût calculatoire. La performance odométrique mesure la capacité du robot à estimer sa position. Dans notre cas, un robot hexapode, sa vitesse désirée v_x, v_y, w_z reste le fruit d'une consigne de vitesse par le contrôleur cinématique de l'hexapode (estimation) issue des calculs de l'espace de travail et du "duty factor". Le robot peut signaler sa position (x, y) dans un système de coordonnées cartésien et également pour indiquer le cap actuel du robot. Le procédé de navigation est composé de plusieurs sous-systèmes permettant au robot de se repérer dans l'espace dans le référentiel carte (AMCL), de connaître son vecteur de déplacement (odométrie avec la correspondance de scans laser), et de déterminer sa trajectoire en fonction d'un coût (algorithme Dijkstra [199]).

3.4 Brique d'asservissement visuel

L'asservissement visuel possède plusieurs entrées principales : le flux vidéo, la vitesse et la localisation du robot. Lors du traitement, le flux vidéo est analysé pour détecter le point d'intérêt et se diriger automatiquement vers celui-ci. Une fois détectées, nous procédons à une identification de l'emplacement de l'objet et une comparaison est réalisée par rapport à l'emplacement de l'objet souhaité. La vitesse du robot est modifiée automatiquement pour atteindre l'objectif. Cette différence produit le déplacement visuel souhaité dans l'image. D'une information visuelle, elle est transformée en information mécanique pour réaliser le déplacement physique souhaité. Il est alors traduit en point de navigation et sera traité par la brique de navigation. L'approche visuelle est composée d'une approche par marqueur QR Code et l'autre d'une approche par réseau de neurones convolutifs. L'objectif de ces deux méthodes est de calculer un point de navigation à partir d'un point d'intérêt. L'approche visuelle par QR Code fonctionne sous le principe d'un marqueur visuel pouvant être identifié par une fonction mathématique et par vision par ordinateur. Le décryptage du marqueur permet d'avoir un point de repère spatial, un ancrage, dans le repère "carte". L'approche visuelle en utilisant un réseau de neurones convolutifs possède plusieurs avantages. Elle permet d'utiliser une inférence sur n'importe quel objet à détecter (en utilisant une boîte englobante pour détecter l'objet) et d'utiliser un jacobien visuel, une opération matricielle pour diriger automatiquement le robot vers l'objet en question.

3.4.1 Asservissement visuel d'un robot hexapode

L'asservissement visuel d'un robot hexapode a pour objectif de le diriger vers un percept visuel. Dans des conditions d'éclairage, de sol et d'environnements dégradés, le procédé d'asservissement visuel peut-être perturbé par le bruit de l'image et induire des erreurs dans la commande. L'asservissement visuel donne la capacité à un robot de se déplacer vers un stimulus visuel de manière automatique. Ce procédé a été majoritairement utilisé dans un but industriel, afin de déplacer des pièces par identification visuelle avec une loi de commande simple [200]).

Le procédé d'une itération de contrôle est obtenu sur 3 étapes. L'acquisition de l'image est la première étape. Une qualité d'image permettra d'avoir une meilleure précision visuelle. La deuxième étape est l'extraction des mesures utiles de l'image en fonction d'un ou plusieurs points de repère. La troisième étape est le calcul des valeurs actuelles des caractéristiques visuelles utilisées comme entrée pour un schéma de contrôle. Une fois les

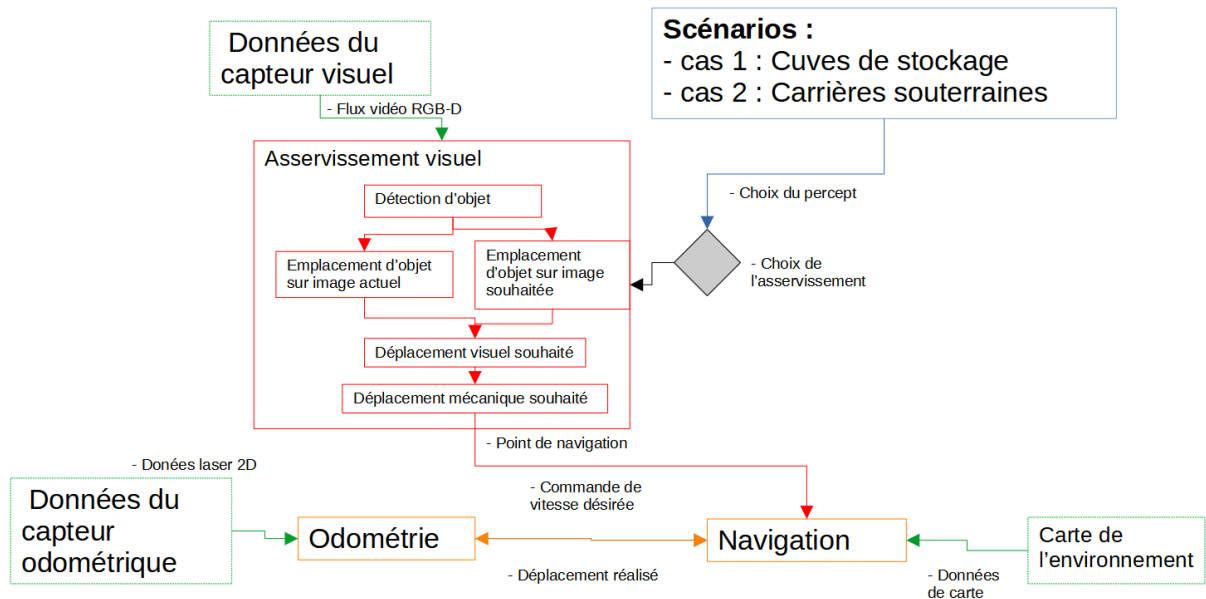


FIGURE 3.7 – Méthode d'asservissement visuel. En rose : la brique d'asservissement visuel, en jaune : la brique de navigation et en vert : les données des capteurs.

valeurs actuelles traitées, un calcul d'erreur s'effectue entre les valeurs actuelles et désirées des caractéristiques visuelles. Enfin, une mise à jour des sorties de contrôles est opérée pour donner un vecteur vitesse au robot.

Théorie

La majorité des tâches d'asservissement visuel peut être modélisée sous la forme d'un procédé de réduction d'une erreur noté $e(t)$ [139] :

$$e(t) = s(m(r(t)), a) - s^*(t) \quad (3.24)$$

Le vecteur $m(r(t))$ est un ensemble de mesures sur l'image (coordonnées des points intérêts). Les coordonnées peuvent être une zone, le centre de gravité ou tout autre point de repère sur l'objet. La position $r(t)$ représente la position de la caméra par rapport à l'environnement. a est une suite de paramètres qui représente des données additionnelles sur le système. Le vecteur $s^*(t)$ contient les valeurs désirées des caractéristiques, constantes sur un cas statique ou variables sur un cas dynamique.

Dans notre cas, le robot hexapode possède la caméra sur sa base. C'est donc un asservissement visuel basé sur l'image ou IBVS. De manière générale, les approches IBVS sont

appliquées sur des bras robotiques articulés où le schéma de contrôle prend en compte la vitesse de la fin d'effecteur q afin d'obtenir la relation entre la variation des caractéristiques dans le temps t et \dot{s} les contrôles d'entrées du robot [139].

$$\dot{s} = J_s \dot{q} + \frac{\delta s}{\delta t} \quad (3.25)$$

Où J_s est la matrice jacobienne des caractéristiques.

Le calcul de la matrice Jacobienne nous donne alors :

$$J_s = L_s V_n^c J(q) \quad (3.26)$$

Où $J(q)$ est le Jacobien du robot tel que $v_n = J(q)\dot{q}$. v_n est la vitesse de la fin d'effecteur. V_n^c est la matrice de transformation du mouvement provenant du capteur sur la fin d'effecteur. Dans notre cas, on considère la fin d'effecteur comme le centre du corps de l'hexapode avec un "offset" pour atteindre la caméra.

La transformation homogène est codée par les paramètres extrinsèques R et t et représente le changement de base du système de coordonnées monde w au système de coordonnées caméra c . Ainsi, étant donné la représentation du point P en coordonnées universelles, P_w , on obtient la représentation de P dans le système de coordonnées de la caméra, P_c , par :

$$P_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} P_w \quad (3.27)$$

Cette transformation homogène est composée de R , une matrice de rotation 3x3, et t , un vecteur de translation 3x1 :

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.28)$$

En combinant la transformation projective et la transformation homogène, nous obtenons la transformation projective qui cartographie les points 3D en coordonnées mondiales en points 2D dans le plan image et en coordonnées caméra normalisées :

$$Z_c \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.29)$$

Asservissement visuel par QRcode

Un marqueur aruco est un marqueur carré synthétique [192], [193]. Il est composé d'une matrice binaire interne et d'une large bordure noire. La taille de la matrice interne est déterminée par la taille du marqueur. Par exemple, une taille de marqueur de 4×4 est composée de 16 bits. Son identifiant (ID) est déterminé par la matrice binaire interne. Il y a de nombreux avantages à utiliser ce marqueur. Par exemple, la bordure noire facilite sa détection rapide dans l'image. Il permet également des techniques de détection et de correction d'erreurs dues à la codification binaire. Dans cette section, un algorithme de détection sera introduit pour détecter et calculer la pose dans un espace 3D à partir des travaux de Ramirez et al. [192]. Les marqueurs peuvent être utilisés comme points de repère 3D pour l'estimation de la pose de la caméra. Nous désignons la taille du marqueur une fois qu'il est imprimé sur un morceau de papier. L'image suivante 3.8 montre le système de coordonnées utilisé dans la bibliothèque aruco.

La figure 3.8 ci-dessous montre le système de coordonnées d'un marqueur de repère. Les coordonnées de l'image (x, y) de chaque sommet correspondent à un rayon de la caméra. Le code d'estimation de pose résout un ensemble d'équations linéaires pour déterminer les coordonnées du repère global (X, Y, Z) de chacun des sommets. À partir de là, nous obtenons la transformation du système de coordonnées du repère vers le système de coordonnées de la caméra T_{fidcam} . Cela représente la pose du marqueur dans le système de coordonnées de la caméra. Dans le diagramme ci-dessous, deux repères, fid_1 et fid_2 sont affichés. Si fid_1 est à une pose connue dans le monde, $T_{mapfid1}$ et que nous connaissons les transformations du marqueur en caméra pour les deux marqueurs, nous pouvons calculer la pose de fid_2 ainsi :

$$T_{mapfid2} = T_{mapfid1} \times T_{camfid2} \times T_{fid1cam} \quad (3.30)$$

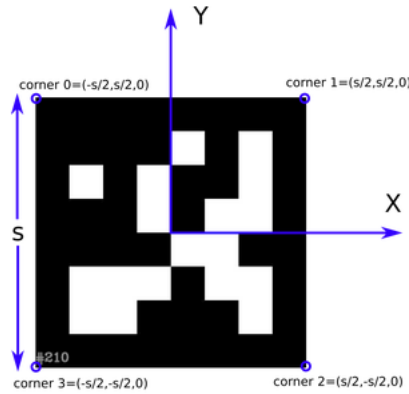


FIGURE 3.8 – Principe de fonctionnement du QR code.

Jacobien d’image

Le Jacobien d’image répond à la problématique de la correspondance de repères locaux entre la caméra (IBVS) et l’objet.

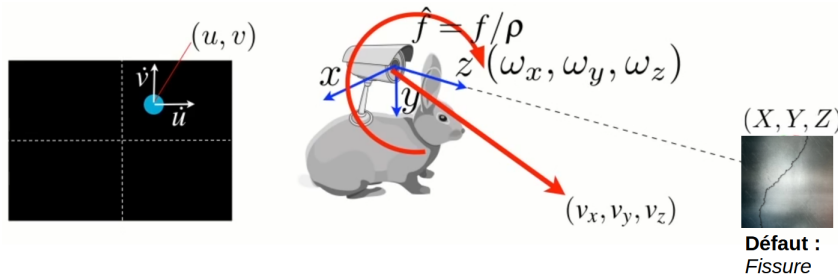


FIGURE 3.9 – Jacobien d’image [140]

Sur la figure 3.9, nous avons une caméra regardant un objet, projetant l’image. Nous attachons une coordonnée C à la caméra et nous avons la coordonnée de l’objet à identifier exprimée par rapport au cadre de coordonnées de la caméra. La distance focale d’un objectif \hat{f} , en mètres, est divisée par les dimensions du pixel données par ρ . ρ s’exprime en mètres par pixel.

Nous pouvons projeter la position d’un objet dans le plan image que nous représentons par les coordonnées u, v . Nous pouvons simplifier la notation et faire référence à la position de l’objet dans l’espace 3D comme X, Y, Z . Nous pouvons écrire les coordonnées de la projection du défaut (fissure) dans le plan image de cette manière, en utilisant les symboles X, Y et Z et \hat{f} , qui rappellent la distance focale de l’objectif exprimée en unités

de pixels. Considérons que la caméra est en mouvement par le biais du robot, la caméra se déplace avec une vitesse. Les coordonnées du vecteur vitesse sont exprimées par trois vecteurs V_x , V_y et V_z et par trois vecteurs représentant la vitesse angulaire ω_X , ω_Y et ω_Z .

Nous prenons l'hypothèse que l'objet (un défaut visuel, par exemple) est projeté dans le plan image à emplacement u , v , avec une vitesse dans le plan image, que nous représenterons par \dot{u} et \dot{v} . L'expression qui relie la vitesse de la caméra et la vitesse du pixel dans le plan image sont liées par une matrice jacobienne. La matrice Jacobienne est ici une matrice 2×6 , qui concerne la vitesse de la caméra et c'est la vitesse spatiale de la caméra qui a une composante de translation et une composante de rotation. Multiplié par cette matrice 2×6 , pour avoir la vitesse dans le plan image en fonction de la vitesse à laquelle ce point de l'image semble se déplacer. La perspective de l'image peut être calculée en projetant l'objet sur l'image (2D) en abscisse :

$$u = \frac{\hat{f} \cdot X}{Z} \quad (3.31)$$

La perspective de l'image peut être calculée en projetant l'objet sur l'image (2D) en ordonnée :

$$v = \frac{\hat{f} \cdot Y}{Z} \quad (3.32)$$

$$\hat{f} = \frac{f}{\rho} \quad (3.33)$$

Avec : (u, v) : Coordonnée de l'objet dans l'image.

(\dot{u}, \dot{v}) : Vitesse de pixel de l'objet dans l'image.

C : Caméra.

(V_x, V_y, V_z) : Translation sur l'axe x,y,z de la caméra.

$(\omega_x, \omega_y, \omega_z)$: Rotation sur l'axe x,y,z de la caméra.

(C_x, C_y, C_z) : Coordonnée de l'objet exprimée en fonction du repère de la caméra.

(X, Y, Z) : Coordonnée de l'objet.

\hat{f} : Taille de focal de la lentille.

ρ : Dimension des pixels (m/pixel), unité de pixel.

Le flux optique est exprimé par \dot{u}, \dot{v} indiquant la vitesse de pixel dans l'image. La

vitesse de pixel est donnée par le jacobien d'image , figure 3.9, en fonction de la vitesse de la caméra sur les différents axes de translation et rotation 3.34.

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \left(\frac{-\hat{f}}{Z}\right) & 0 & \frac{u}{Z} & \frac{uv}{\hat{f}} & \left(-\frac{\hat{f}+u^2}{\hat{f}}\right) & v \\ 0 & -\left(\frac{\hat{f}}{Z}\right) & \frac{v}{Z} & \frac{\hat{f}+v^2}{\hat{f}} & \left(\frac{-uv}{\hat{f}}\right) & -u \end{pmatrix} \times \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (3.34)$$

Dans l'équation (3.34), le Jacobien d'image est réalisé pour un point, une coordonnée x et y sur une image plane, noté $J_p(u, v, Z)$.

$$J_p(u, v, Z) = \begin{pmatrix} \left(\frac{-\hat{f}}{Z}\right) & 0 & \frac{u}{Z} & \frac{uv}{\hat{f}} & \left(-\frac{\hat{f}+u^2}{\hat{f}}\right) & v \\ 0 & -\left(\frac{\hat{f}}{Z}\right) & \frac{v}{Z} & \frac{\hat{f}+v^2}{\hat{f}} & \left(\frac{-uv}{\hat{f}}\right) & -u \end{pmatrix} \quad (3.35)$$

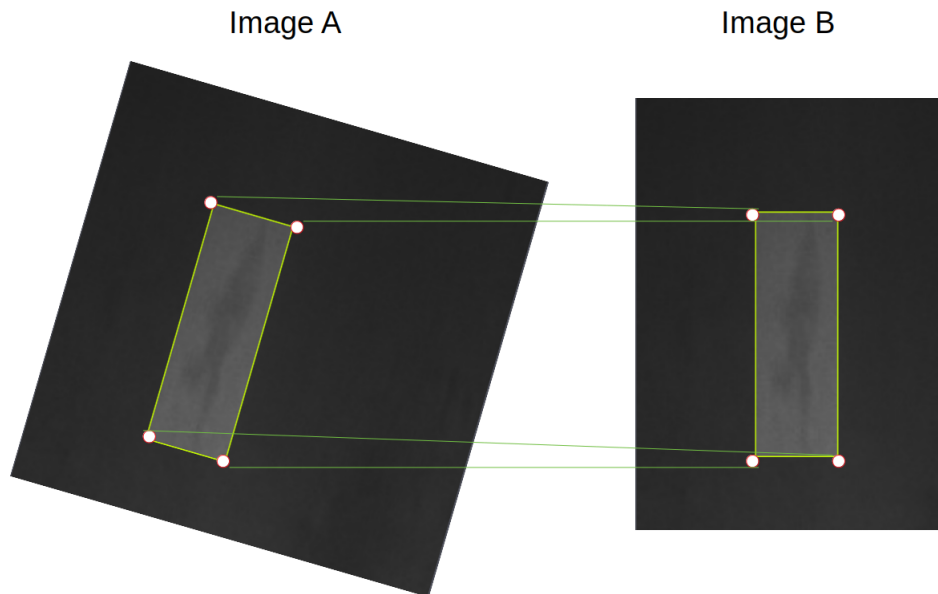


FIGURE 3.10 – L'image A représente un détection automatique avec une boîte englobante de l'objet détecté en vue initial et l'image B représente la vue désirée du même objet (défaut industriel).

Mouvement sur 4 points

Si nous avons par exemple, figure 3.10, une boîte englobante d'un algorithme de détection d'objet automatique, nous devons considérer cette approche sur 4 points. Nous avons la vue initiale de l'objet. Dans ce cas, notre objet est un rectangle et défini par les coordonnées de ses 4 coins.

Nous pouvons donc définir notre problème de changer la vue de l'objet d'une vue initiale à une vue désirée. L'objectif est de faire correspondre notre vue de l'objet désirée avec notre vue initiale de l'objet. En définitive, on souhaite déplacer les 4 points (cercles) de l'image dans la disposition de l'image B, figure 3.10. Pour se déplacer, ces points doivent avoir une certaine vitesse vers les points désirés correspondant. La relation entre la vitesse des pixels et la vitesse de la caméra est donné par la matrice jacobienne d'image (3.36). On peut écrire la matrice sous forme de fonction Jp et cela signifie que le jacobien pour une entité ponctuelle est une fonction de la coordonnée du pixel, u et v . Il indique à quelle distance le point est éloigné dans le monde, qui est donné par Z .

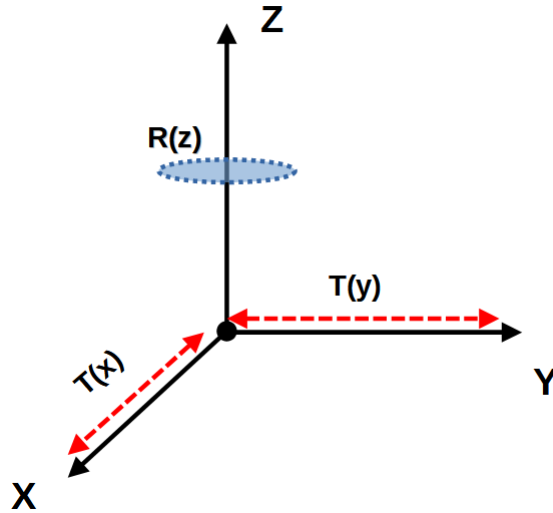


FIGURE 3.11 – Déplacement d'un robot holonome dans un repère cartésien.

$$\begin{pmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \\ \dot{u}_4 \\ \dot{v}_4 \end{pmatrix} = \begin{pmatrix} J_{p1}(u_1, v_1, Z_1) \\ J_{p2}(u_2, v_2, Z_2) \\ J_{p3}(u_3, v_3, Z_3) \\ J_{p4}(u_4, v_4, Z_4) \end{pmatrix} \times \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (3.36)$$

Le robot hexapode peut se déplacer de manière holonome, il est capable de se déplacer dans n'importe quelle direction, quelle que soit son orientation, Figure 3.11. Ce déplacement est caractérisé par une translation selon x permettant d'avancer ou de reculer, une translation selon y pour les déplacements à gauche ou à droite, une rotation selon z permettant de tourner vers la gauche ou la droite.

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.37)$$

On remplace ainsi les termes dans l'équation. On a donc :

$$\begin{pmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \\ \dot{u}_4 \\ \dot{v}_4 \end{pmatrix} = \begin{pmatrix} J_{p1}(u_1, v_1, Z_1) \\ J_{p2}(u_2, v_2, Z_2) \\ J_{p3}(u_3, v_3, Z_3) \\ J_{p4}(u_4, v_4, Z_4) \end{pmatrix} \times \begin{pmatrix} v_x \\ v_y \\ v_z = 0 \\ \omega_x = 0 \\ \omega_y = 0 \\ \omega_z \end{pmatrix} \quad (3.38)$$

$$V = J\dot{\theta} \quad (3.39)$$

$$J^T V = J^T J \dot{\theta} \quad (3.40)$$

$$(J^T J)^{-1} J^T V = (J^T J)^{-1} J^T J \dot{\theta} \quad (3.41)$$

La méthode de la pseudo-inverse Monroe-Penrose s'écrit alors :

$$J^+ = \dot{\theta}/V \quad (3.42)$$

Le cas d'application sur 4 et 3 points peuvent être résolue par la pseudo-inverse.

Cas d'application sur 3 points

Le cas le plus simple pour un asservissement visuel est le cas d'un asservissement sur 3 points.

$$\begin{pmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \end{pmatrix} = \begin{pmatrix} J_{p1}(u_1, v_1, Z_1) \\ J_{p2}(u_2, v_2, Z_2) \\ J_{p3}(u_3, v_3, Z_3) \end{pmatrix} \times \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (3.43)$$

Le calcul du jacobien nécessite le calcul des déterminants pour définir si la matrice est inversible. Le calcul du déterminant pour une matrice A d'ordre n peut être réalisé suivant la formule suivante :

$$\det(A) = \sum_{i=1}^n (-1)^{i-1} \times \det(A_{A_i}) \times a_{i1} \quad (3.44)$$

Où A_i est la matrice carrée d'ordre $n-1$ obtenue en supprimant la première colonne et la i -ème ligne de A. Le calcul du déterminant sur une matrice 6×6 est donc réalisé selon :

$$\begin{vmatrix} + & - & + & - & + & - \\ - & + & - & + & - & + \\ + & - & + & - & + & - \\ - & + & - & + & - & + \\ + & - & + & - & + & - \\ - & + & - & + & - & + \end{vmatrix} \quad (3.45)$$

$$\det(J_{\Sigma p}) = \det \begin{pmatrix} \left| J_{p1}(u_1, v_1, Z_1) \right| \\ \left| J_{p2}(u_2, v_2, Z_2) \right| \\ \left| J_{p3}(u_3, v_3, Z_3) \right| \end{pmatrix} \quad (3.46)$$

Étant donné que la matrice Jacobienne est de dimension 6×6 , elle est donc carrée. Il est alors possible de l'inverser. Ainsi, nous pouvons décomposer les déterminants selon la formule de Leibniz. Le déterminant d'une matrice carrée, notée $A = (a_{ij})$ d'ordre n est :

$$\det(A) := \sum \epsilon(\sigma) a_{i, \sigma(i)} \quad (3.47)$$

Où S_n est le groupe des permutations et pour une permutation σ de S_n , $\epsilon(\sigma)$ désigne

sa signature, égale à 1 si la permutation est paire et -1 dans le cas contraire.

Ainsi les déterminants sont calculés selon cette méthode [141]. Notons $\dot{y}(q)$ est la vitesse des points sur l'image, $\dot{q}(t)$ est la vitesse du robot, J représente la matrice d'image Jacobienne 3.48.

$$\dot{y}(q) = J \cdot \dot{q}(t) \quad (3.48)$$

L'erreur liée est notée $f(q, t)$ et est une fonction non linéaire 3.49 :

$$f(q, t) = y(q) - \dot{y}(t) \quad (3.49)$$

Pour le cas d'un asservissement visuel sur 4 points, l'inversion de matrice ne peut être réalisée qu'à partir d'une matrice carrée. Or, dans le cas d'un asservissement sur 4 points, nos dimensions de matrices sont de la forme :

$$\dot{y}(q)_{[6 \times 1]} = J_{[8 \times 6]} \cdot \dot{q}(t)_{[6 \times 1]} \quad (3.50)$$

Le Jacobien d'image n'étant pas en matrice carré 3.51, l'inversion par la méthode Leibniz [142] n'est pas réalisable. Une solution simple peut être de sélectionner les vitesses non-pertinentes (comme la vitesse angulaire sur X et Y) 3.51 dans le vecteur vitesse et ainsi de simplifier l'équation. Cette méthode fonctionne surtout pour une approche de Jacobien mécanique avec 4 effecteurs de type rotoïde ou tout autre configuration où le nombre de points d'application N est inférieur à 6. Un exemple pour 4 points d'application est noté :

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} J_{1,1} & J_{1,2} & J_{1,3} & J_{1,4} \\ J_{2,1} & J_{2,2} & J_{2,3} & J_{2,4} \\ J_{3,1} & J_{3,2} & J_{3,3} & J_{3,4} \\ J_{4,1} & J_{4,2} & J_{4,3} & J_{4,4} \\ J_{5,1} & J_{5,2} & J_{5,3} & J_{5,4} \\ J_{6,1} & J_{6,2} & J_{6,3} & J_{6,4} \end{pmatrix} \times \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} \quad (3.51)$$

Si les valeurs de vitesse ω_x et ω_y ne sont pas significatives, l'expression peut alors être simplifiée. La matrice devient alors inversible 3.52.

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_z \end{pmatrix} = \begin{pmatrix} J_{1,1} & J_{1,2} & J_{1,3} & J_{1,4} \\ J_{2,1} & J_{2,2} & J_{2,3} & J_{2,4} \\ J_{3,1} & J_{3,2} & J_{3,3} & J_{3,4} \\ J_{6,1} & J_{6,2} & J_{6,3} & J_{6,4} \end{pmatrix} \times \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} \quad (3.52)$$

Une approche classique est de réaliser une pseudo-inverse Moore-Penrose [143] dans les cas des matrices carrées non-inversibles, mais généralisable sur des matrices non-carrées à coefficients réels ou complexes. Considérons une matrice A avec n lignes et p colonnes dont la pseudo-inverse est notée A^+ et la matrice adjointe notée A^* vérifiant les conditions suivantes dans les équations 3.53, 3.54, 3.55 et 3.56.

$$AA^+A = A \quad (3.53)$$

$$A^+AA^+ = A^+ \quad (3.54)$$

$$(AA^+)^* = AA^+ \quad (3.55)$$

$$(A^+A)^* = A^+A \quad (3.56)$$

Dans le calcul, on considère notre même matrice A de dimension $n \times p$, de rang k . On peut alors considérer un produit de matrices tel que $A = BC$, avec B possédant n lignes et k colonnes et C de k lignes et p colonnes. Les produits (CC^*) et (BB^*) sont inversibles et la relation est vérifiée :

$$A^+ = C^*(CC^*)^{-1}(B^*B)^{-1}B^* \quad (3.57)$$

Cette méthode estime la matrice pseudo-inverse du Jacobien d'image indépendamment du type du robot, le type de système d'asservissement visuel et évite la singularité de la matrice Jacobienne. Le résultat de la méthode pseudo-inverse sur le jacobien visuel est :

$$J^+ = (J^T J)^{-1} J^T \quad (3.58)$$

Dans cette partie, nous avons évoqué l'aspect géométrique et mathématique que peut impliquer l'asservissement visuel d'un robot hexapode. L'odométrie est une source clé

d'information pour tout le reste du processus d'automatisation. La qualité de la caméra a une importance sur la qualité d'asservissement visuel, car elle sert à extraire les caractéristiques visuelles pour la loi de commande. L'asservissement visuel peut être réalisé de différentes façons : l'utilisation d'un QRcode permet d'obtenir les informations de positions et orientations par traitement numérique. C'est un moyen efficace pour tester ou calibrer un programme d'asservissement visuel et de voir les réactions en simulation et sur le robot réel. D'autres méthodes utilisent la détection automatique d'objets et les repères se font à partir de la boîte englobante.

3.4.2 La détection automatisée de points d'intérêts

Nous avons vu précédemment, dans le chapitre 2, l'intérêt d'utiliser des algorithmes d'apprentissage pour les percepts visuels. Les réseaux de neurones convolutifs permettent de détecter des objets grâce à une boîte englobante à partir d'une base de données labellisées. Ces algorithmes peuvent opérer en temps réel sur des architectures possédant des capacités de calculs en temps réel par l'utilisation de GPU. Les types de défauts peuvent être ainsi traités en tant que classe d'objet.

Les réseaux de neurones convolutifs de région

La différence entre les algorithmes de détection d'objets et les algorithmes de classification [202] est que dans les algorithmes de détection, nous essayons de tracer une boîte englobante autour de l'objet d'intérêt pour le localiser dans l'image. De plus, la boîte englobante n'est peut-être pas nécessairement un seul cadre de délimitation dans un cas de détection d'objets, il pourrait y avoir de nombreux cadres de délimitation représentant différents objets d'intérêt dans l'image et sans savoir combien il y aura d'objets à l'avance.

La principale raison pour laquelle on ne peut pas résoudre ce problème en construisant un réseau convolutif standard suivi d'une couche entièrement connectée est que la longueur de la couche de sortie est variable car le nombre d'occurrences des objets d'intérêt n'est pas fixe. Une approche naïve pour résoudre ce problème serait de prendre différentes régions d'intérêt de l'image et d'utiliser un CNN (réseau de neurones convolutif) pour classer la présence de l'objet dans cette région. Le problème avec cette approche est que les objets d'intérêt peuvent avoir des emplacements spatiaux différents dans l'image et des rapports d'aspect différents. Par conséquent, il faudrait sélectionner un grand nombre de régions et

R-CNN avec des caractéristiques CNN

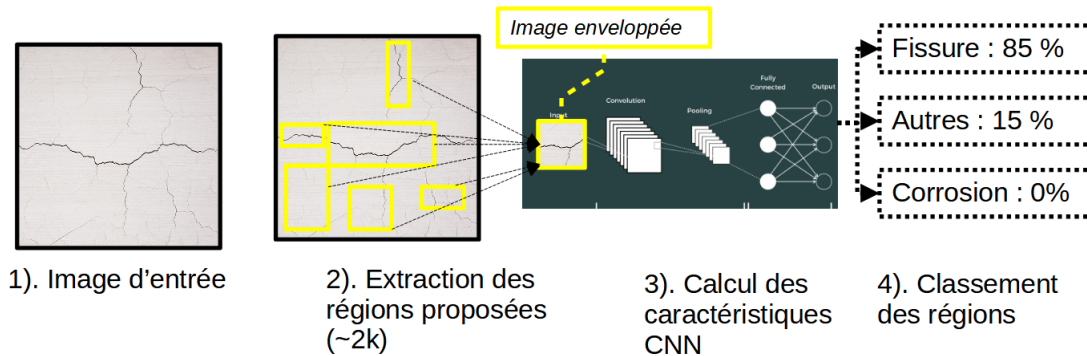


FIGURE 3.12 – Les images d'entrées sont découpées en régions où un objet est susceptible d'apparaître (extraction des caractéristiques). Ces régions sont ensuite classées par le réseau de neurones convolutif [203].

cela pourrait exploser en termes de calcul. Par conséquent, des algorithmes comme SSD ou YOLO ont été développés pour trouver ces occurrences et les trouver rapidement. Afin de procéder à la détection automatique des défauts, nous choisissons un algorithme de type "R-CNN", un réseau de neurones convolutifs qui a la particularité de pré-découper l'image en régions dans le but d'accélérer le traitement.

Nous avons présenté ici deux approches d'asservissement visuel. Une approche à partir d'un QR code a été éprouvée en simulation. Et l'autre à partir d'un réseau de neurones convolutif (YOLO v5), pour un défaut industriel à détecter.

3.5 Brique d'asservissement chimique

L'approche utilisée est décrite Figure 3.13. La méthode d'asservissement chimique utilise les données des capteurs de gaz et les données de l'anémomètre. L'algorithme calcule des points de navigation par itération dans le but de remonter le panache de gaz jusqu'à la source. Les points de navigation sont traités par la brique de navigation pour orienter le robot dans la bonne direction grâce à l'odométrie et à la brique de localisation AMCL dans le repère carte. L'asservissement chimique possède une entrée principale : celui du flux chimique, composée de l'information de concentration du gaz (en ppm) et de la direction du vent et celui de localisation du robot. Ces données transitent dans l'algorithme de recherche de gaz pour produire des points de navigation par itération.

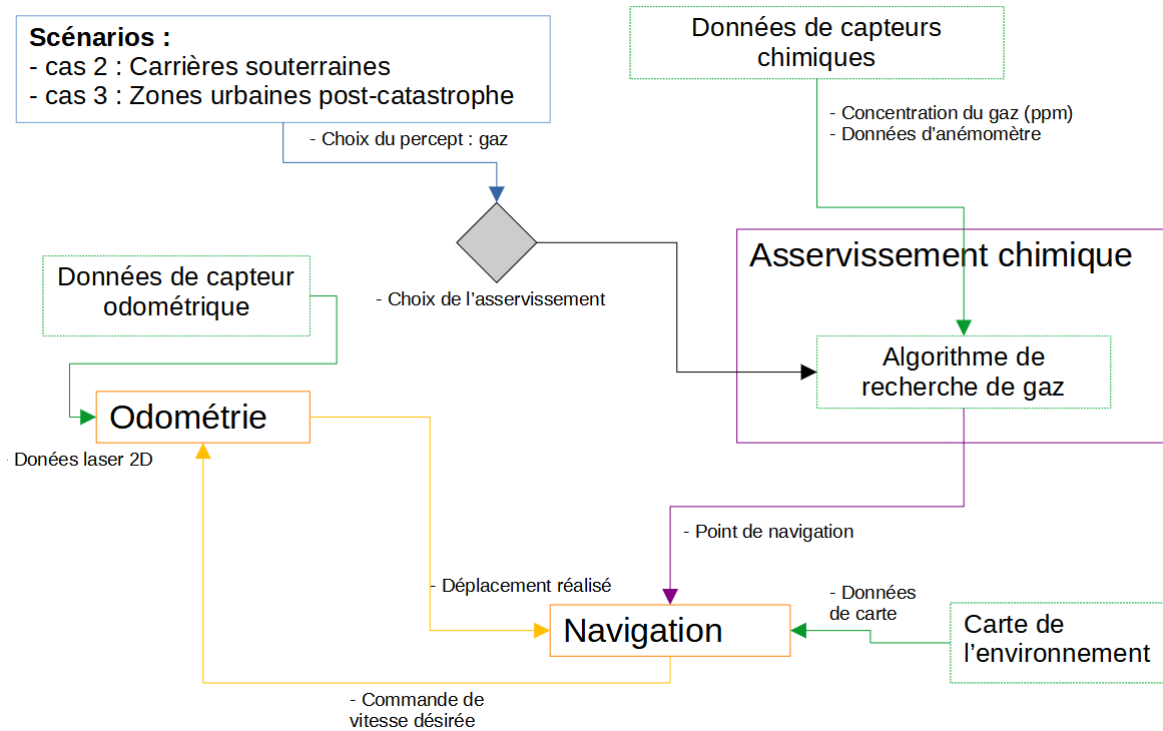


FIGURE 3.13 – Méthode d'asservissement chimique. En violet : la brique d'asservissement chimique, en jaune : la brique de navigation et en vert : les données des capteurs.

3.5.1 Asservissement chimique d'un robot hexapode

L'asservissement chimique d'un robot hexapode implique d'avoir à appréhender la diffusion du gaz pour en rechercher la source. Dans cette partie, nous allons aborder les notions nécessaires pour réaliser un asservissement chimique, tant sur les capteurs que sur la partie algorithme. L'asservissement chimique comporte des problématiques : la modélisation d'un flux de gaz, la détection et la recherche d'une fuite de gaz. Les environnements pour notre cas 1 et cas 2 impliquent un risque de fuite de gaz par le biais de canalisation. La modélisation d'un flux de gaz est un problème complexe où il faut créer des cartes de distributions de gaz, planifier une couverture complète de tournées de surveillance, discriminer entre les différents volatiles ou déduire l'emplacement de la fuite de gaz. Dans un environnement ouvert infini, une fuite de gaz diffuse avec un certain coefficient de diffusion [196]. L'appréhension des fuites de gaz s'exprime en utilisant le calcul de Fick. C'est une relation linéaire entre le flux de matière et le gradient de concentration de celle-ci. On a donc :

$$\vec{f} = -k\nabla C \quad (3.59)$$

$$\frac{\delta C}{\delta t} = -\nabla \vec{f} \quad (3.60)$$

Où $C = C(x, y, z; t)$ représente la concentration de gaz à l'emplacement $l = (x, y, z)$ en unités de mg/m^3 , $\vec{f} = \vec{f}(x, y, z; t)$ est le flux de diffusion, en unités de mg/m^3 , et k est le coefficient de diffusion, en unités de m^2/s . Le flux de diffusion est noté $\vec{f} = \vec{f}(x, y, z; t)$ en mg/m^3 . Les travaux de Li et al., 2015 [196] déterminent que l'équation de diffusion idéale est notée :

$$\frac{\delta C}{\delta t} = k\nabla^2 C - \vec{v}\nabla C \quad (3.61)$$

Où le vecteur \vec{v} est la vitesse naturelle du vent, en m/s .

Dans un scénario ouvert, les équations sont applicables et la force du vent peut s'appliquer. Dans un scénario à environnement fermé, on considère alors que $\vec{v} = 0$. La problématique d'une fuite de gaz dans un tuyau est de pouvoir définir la diffusion à partir d'une perte de pression dans le réservoir ou le tuyau. La fuite de gaz peut être calculée en termes de perte de pression dans le réservoir [144].

$$L = -V \frac{dP}{dt} \quad (3.62)$$

Où L est le taux de fuite (Pam^3/s), P est la pression (Pa), V est le volume du réservoir (m^3), et t est le temps en seconde.

Étant donné que le taux de fuite dépend de la différence entre la pression du réservoir et la pression atmosphérique, cela doit être spécifié lors du calcul des taux de fuite. Lorsque le taux de fuite est pour une installation sous vide et que la différence n'est pas spécifiée, l'hypothèse est que la différence de pression est de 1 atm.

Le taux de fuite pour des gaz différents à travers le même canal est du à la variation de la masse des gaz (M) et leurs volumes pendant l'expansion (V). Le flux d'un gaz peut être une fonction notée :

$$\sqrt{\frac{V}{M}} \quad (3.63)$$

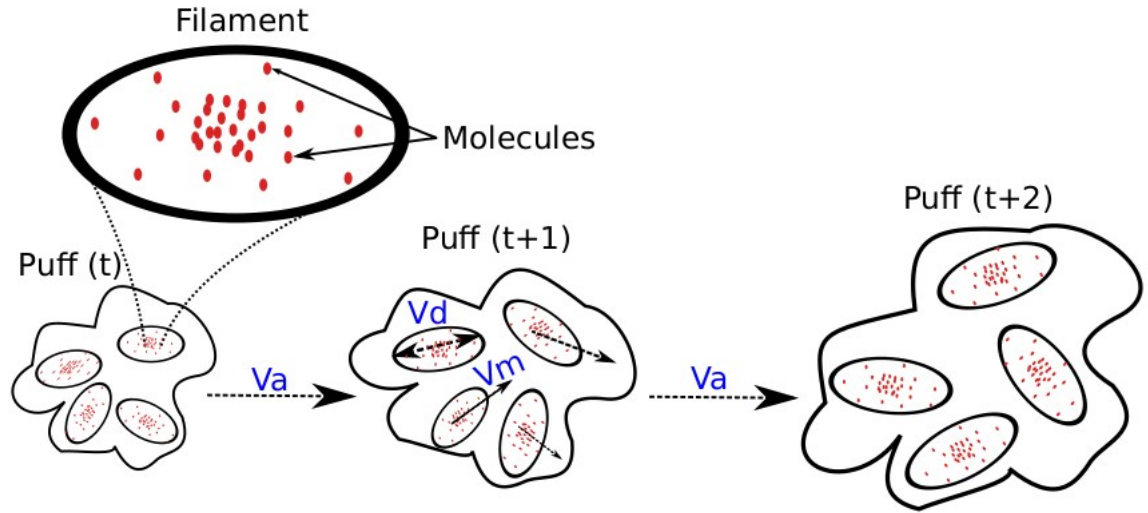


FIGURE 3.14 – Un dégagement de gaz est modélisé comme une séquence de bouffées, chacune est composée de plusieurs filaments (étant un filament une distribution normale 3D des molécules de gaz). Les bouffées sont affectées par l'advection et diffusion, altérant l'emplacement des filaments par l'effet du vent (V_a) et des processus aléatoires (V_m), ainsi que leur taille (V_d) [176].

Le taux de fuite établit la relation entre un gaz et un autre, ainsi que la température :

$$L_1 = L_2 \sqrt{\frac{M_2}{M_1}} \quad (3.64)$$

Où : L_1 est le vrai taux de fuite du gaz 1, L_2 est le vrai taux de fuite de gaz 2, M_1 est le poids de la molécule du gaz 1, M_2 est le poids de la molécule du gaz 2.

Par exemple :

$$L_{\text{azote}} = L_{\text{helium}} \sqrt{\frac{M_{\text{helium}}}{M_{\text{azote}}}} = L_{\text{helium}} \sqrt{\frac{4}{28}} = (0,378) L_{\text{helium}} \quad (3.65)$$

Les travaux de [182] sur l'olfaction robotique ont proposé un cadre de simulation de dispersion de gaz 3D : GADEN3D. Il a été présenté dans [183]. L'étude et la recherche de fuite de gaz à l'aide de robots terrestres impliquent que les gaz à détecter soient plus lourds que l'air.

Les capteurs de gaz en métal oxyde

Les capteurs de gaz en oxyde de métal (Figure 7.1) sont basés sur une variation de résistance (R_S) produit par l'exposition du gaz [146]. Au lieu de travailler directement avec R_S , nous avons utilisé le rapport entre R_S et R_0 (résistance du capteur en air propre) pour compenser les dérives potentielles du capteur. La concentration du gaz C_{gaz} peut alors être calculé suivant R_S et R_0 : $C_{gaz} = R_S/R_0$. Cette dérive peut provenir de changements dans les facteurs environnementaux (par exemple, la température, l'humidité, pression ou débit d'air) ou du capteur lui-même [204].

Les algorithmes de recherche de gaz

Les algorithmes de recherche de gaz permettent de trouver la source de l'émission. Même si la majorité des gaz industriels sont plus lourds que l'air, la principale problématique est la nature aléatoire de l'action de l'environnement sur la propagation et la manière de diffuser le gaz [196]. Comme le montre notre étude (paragraphe 2.5), nous allons utiliser la méthode "grid". La méthode employée pour l'asservissement chimique est une forme de "grid search". C'est essentiellement un algorithme d'optimisation utilisée qui, à partir d'une liste de paramètres que l'on produit, apporte une automatisation par des "essais et erreurs".

L'algorithme "Grid" [208] est utilisé pour réaliser la recherche de source de gaz. Cet algorithme probabiliste s'articule autour de la propagation d'estimations locales dans tout l'environnement. En exploitant la géométrie de l'environnement comme base de cette propagation, on évite de s'appuyer sur des modèles analytiques de dispersion, éliminant le besoin d'assumer un environnement contrôlé. L'algorithme présenté dans ce travail suit la norme d'une structure de GSL basé sur l'estimation. Nous l'avons vu précédemment, la réalisation de la méthode implique différentes briques technologiques : le contrôle de l'hexapode, l'odométrie, la navigation, la détection d'objet, les algorithmes d'asservissement visuel et chimique.

3.6 Conclusion de fin de chapitre

Dans ce chapitre, nous avons présenté différentes méthodes en adéquation face aux contraintes d'environnement des différents cas, voir chapitre 2. La génération de la cinématique inverse permet d'activer la démarche pour franchir des obstacles ou s'adapter

à des situations de sol dégradé. L'emploi d'une méthode d'odométrie adaptée pour les robots hexapodes a été définie. La détection des percepts visuels se fera à l'aide d'un algorithme d'apprentissage automatique par réseau de neurones convolutifs. L'asservissement visuel se fera dans un premier temps sur un point, un QRcode servira de percept visuel, afin d'apporter une preuve de concept et de définir les axes d'amélioration. La détection des percepts chimiques se fera à l'aide d'un capteur électronique en oxyde de métal. L'algorithme de recherche de gaz prendra en compte la concentration en gaz et la vitesse du vent. Les algorithmes énoncés précédemment répondent aux différentes problématiques des cas d'application. Ces algorithmes vont être intégrés en simulation et sur le robot réel sur une architecture de robot hexapode, ArIA Inspector.

IMPLÉMENTATION ET INTÉGRATION

Dans ce chapitre, nous allons décrire les moyens mis en œuvre afin d'intégrer les différentes méthodes dans une architecture robotique embarquée. Les évolutions des différents prototypes du robot développé sont présentées. L'étude pratique évoquera les briques logicielles communes aux méthodes d'asservissement. Les évolutions des prototypes, l'architecture logicielle, l'asservissement visuel et chimique seront détaillés dans ce chapitre.

4.1 Application de l'approche

Dans cette section, nous allons aborder l'architecture du robot, l'architecture logicielle et la partie navigation. Nous présentons l'approche détaillée d'asservissement visuel et chimique en soulignant les différentes parties, voir figure 4.1 (se retrouve aussi en annexe avec uniquement le logigramme 7.2). Dans cette proposition, figure 4.1, nous présentons deux méthodes distinctes pour réaliser l'asservissement visuel et chimique. Cette approche globale est composée de différentes parties : la partie navigation, la partie IA, la partie visuelle et la partie chimique. Nous utilisons une caméra stéréoscopique avec capteur de profondeur et une centrale inertiel (IMU) embarquées. Une caméra sera utilisée pour la détection des percepts visuels et le capteur de profondeur mesurera la distance entre le robot et le percept. L'IMU permettra d'avoir une information odométrique du déplacement de la caméra dans le repère "carte" (monde). Le lidar réalise la cartographie de l'environnement et effectue le calcul de l'odométrie. Les capteurs de gaz en oxyde de métal seront situés sur 3 côtés de l'hexapode et seront utilisés pour obtenir la concentration du gaz. Un anémomètre est aussi nécessaire pour indiquer la direction du flux d'air.

La partie contrôle du robot concerne la géométrie et la cinématique du robot. Cette étape permet de réaliser la démarche du robot en prenant en compte la position des fins d'effecteurs (pattes). La démarche calculée est généralement insuffisante pour fournir une odométrie fiable et nous proposons une méthode pour fournir une odométrie robuste du robot dans un espace contraint. Dans la partie navigation, nous aborderons la partie localisation, cartographie, planification de trajectoire et évitement d'obstacles. Cette brique technologique sera associée à l'asservissement visuel et chimique pour assurer un asservissement avec évitements d'obstacles.

Une première version du contrôle par percept visuel sera réalisée en utilisant un QRcode comme point d'intérêt. Une deuxième version sera axée sur les défauts industriels (les fissures, par exemple). La détection automatique sera réalisée sous forme de boîte englobante par un réseau de neurones convolutifs. Les quatre coins de la boîte englobante serviront de données d'entrées pour le jacobien visuel. Cette dernière étape permettra de générer la loi de commande en se basant sur la vitesse des pixels.

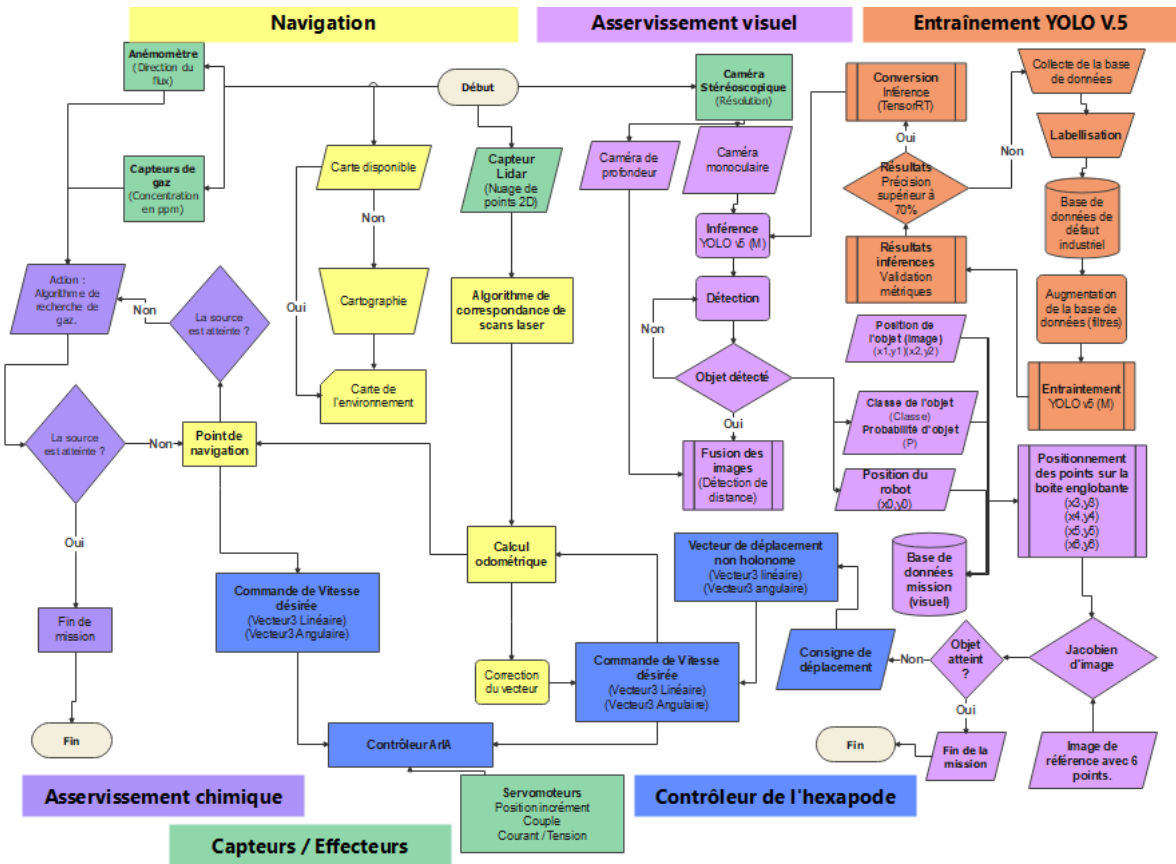


FIGURE 4.1 – Logigramme de l'approche de résolution pour un procédé d'asservissement visuel et chimique

4.1.1 Architecture du robot

L'architecture du robot choisie est celle d'un robot hexapode (6 pattes) afin de mener les différentes opérations de reconnaissance dans des milieux contraints et dégradés. Le robot théorique défini dans le chapitre 2 est composé de pattes inspirées des araignées et d'orientation frontale. Les genoux (fémurs) sont orientés vers l'extérieur et chaque patte sera composée de 3 degrés de liberté. La figure 4.2 illustre les choix d'architecture mécanique du robot.

Nous suggérons l'hypothèse suivante : les efforts sur les coxis sont moins importants que les fémurs et les tibias. Les dimensions du robot sont utilisées pour réaliser le modèle cinématique.

Architecture du robot hexapode.

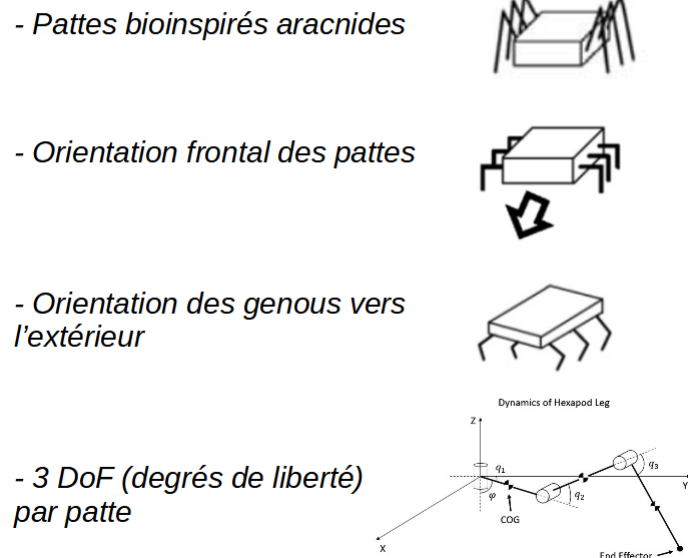


FIGURE 4.2 – Architecture du robot théorique : le robot hexapode.

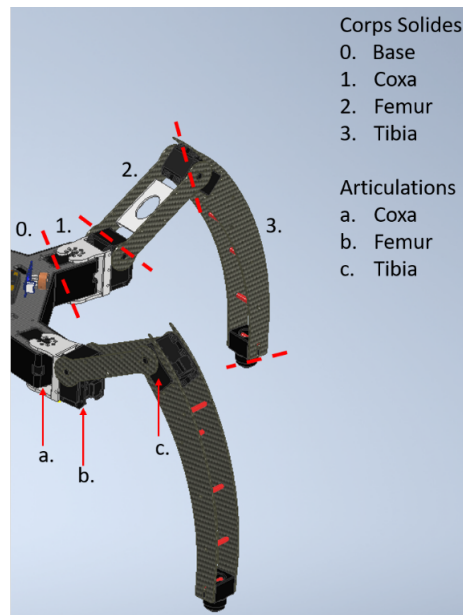


FIGURE 4.3 – Composition d'une patte.

Robot Hexapode : ArIA

Afin de répondre à des besoins client dans le domaine de l'inspection industrielle, plusieurs versions d'ArIA ont été développées. Nous allons présenter ces différentes versions.

ArIA AX18 : 1ère version

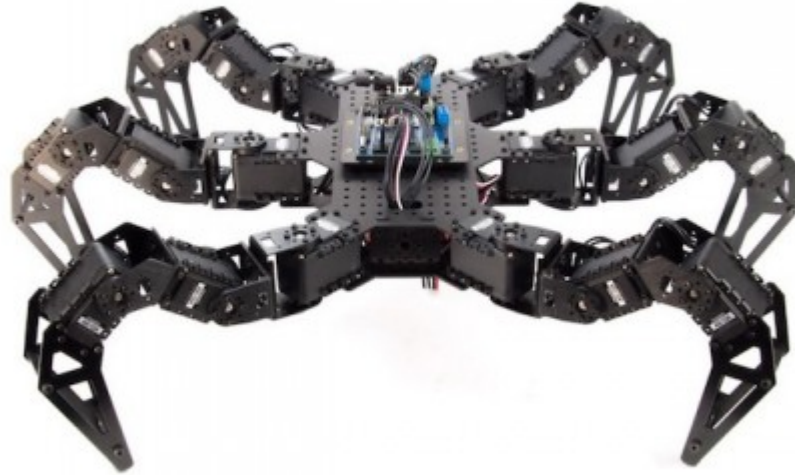


FIGURE 4.4 – ArIA AX18 : 1ère version

La première version d'ArIA est issue d'un châssis constructeur (Figure 4.4) de la marque Trossen Robotics, le modèle "PhantomX Hexapod". Trossen Robotics propose un code source, le "NUKE" ("Nearly Universal Kinematics Engine"). Ce code permet de générer le contrôle cinématique inverse pour le robot et les moteurs. À partir des dimensions du robot (longueur du corps, coxis, fémur et tibia), il génère un code Arduino résolvant la cinématique directe et inverse. Cependant, ce code comporte des limitations : le contrôle du corps n'est pas réalisable et la fluidité du mouvement est perfectible. Le code Phoenix est le code utilisé pour cette version. Sur ce prototype, la taille des coxis est de 52 mm, les fémurs font 66 mm et les tibias font 133 mm. Les moteurs utilisés sont des AX-18 avec une tension de 12 V pour un couple de décrochage de 1,80 Nm, une amplitude d'angle de 300° et un poids unitaire de 55,90 g. La communication des commandes se fait par protocole XBee.

ArIA LUCAN : 2ème version

ArIA LUCAN (Figure 4.5 et 4.6) est la version améliorée en motorisation de ArIA AX18. Dans cette version, le souhait de l'entreprise était d'avoir une version de l'hexapode ayant des caractéristiques permettant aussi l'exploration extérieure.

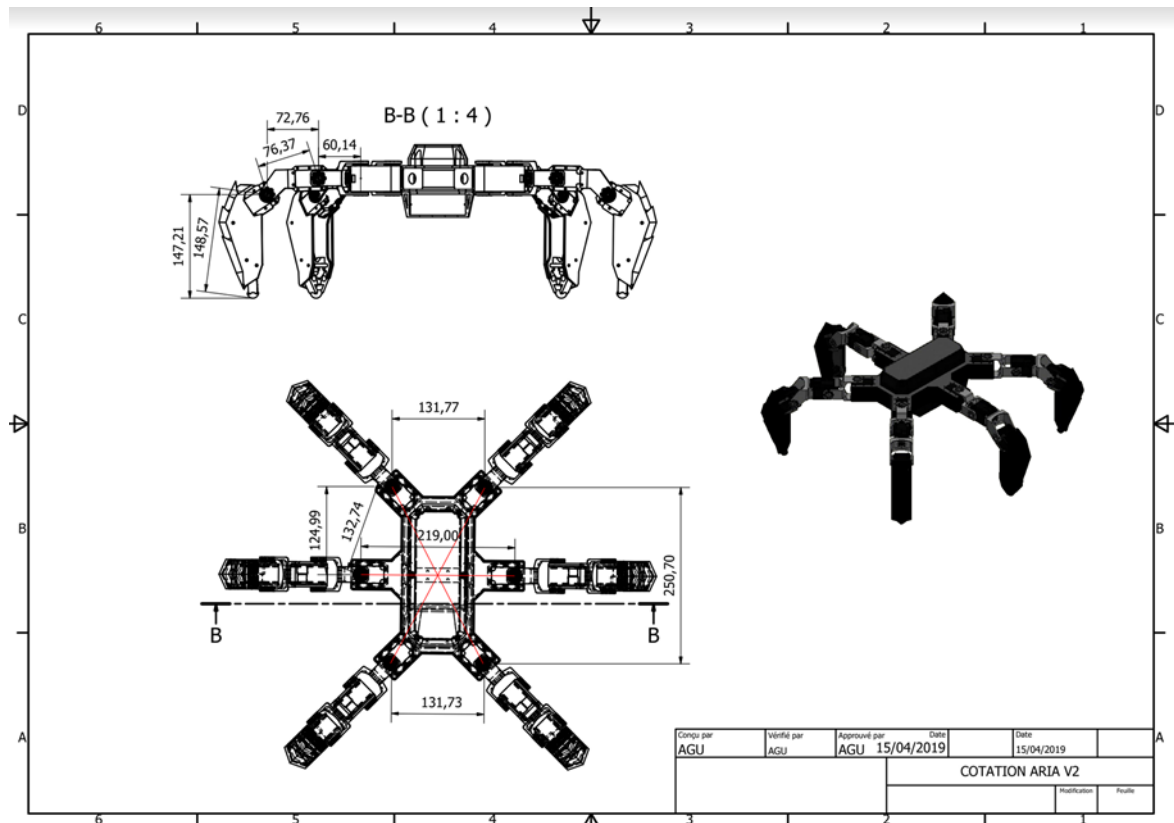


FIGURE 4.5 – ArIA LUCAN, schéma de conception de la deuxième version.

Le corps et les segments des pattes ont été agrandis d'environ 15% (figure 4.5) et le système comporte un équipement vidéo intégré permettant la rediffusion du signal vidéo sur un écran externe par radio-fréquence. Le contrôle du robot est réalisé par une communication par protocole XBee (portée : environ 60 m). Les servomoteurs utilisés (MX28T) ont un couple moteur de 2.5 $N.m$ pour un poids hors charge utile de 4.2 kg. La charge utile est de 2 kg, permettant l'emport de capteurs et éclairages d'appoints.

Ce produit a été réalisé pour le client MC Clic dans le but de réaliser une levée de doute en cas d'intrusion et dans le cadre d'opérations militaires. Pour les deux versions, ArIA AX18 et ArIA LUCAN, un microcontrôleur (ARBOTIX-M) est installé et réalise



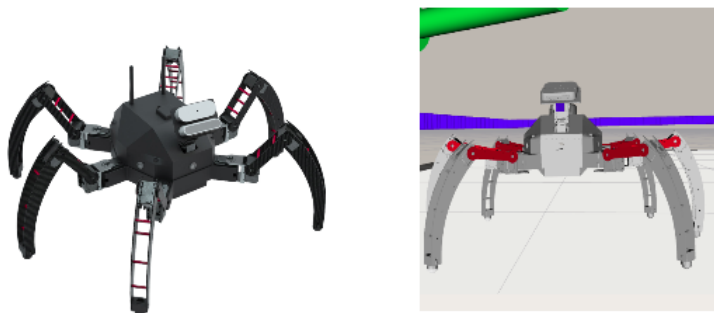
FIGURE 4.6 – ArIA LUCAN.

l'interface avec les servomoteurs.

ArIA INSPECTOR : 3ème version

Le prototype de thèse utilisé est ArIA INSPECTOR. ArIA INSPECTOR (Figure 4.7 et 4.8) est la première version de la gamme à utiliser le système ROS. Sur la figure 4.7, à gauche est présenté le robot réel ArIA INSPECTOR et à droite, la version du modèle dans la simulation GAZEBO. Les propriétés du robot réel, comme le poids des pièces et la géométrie du robot, sont intégrées dans la simulation. Sur la figure 4.8, on peut voir les différentes spécifications du robot.

L'architecture ROS à la particularité d'être développée pour ajouter de la modularité aux architectures robotiques. Cette modularité est basée sur le partage des flux de données à travers des nœuds (les "nodes"), des applications ou exécutables pouvant être codés sur différents langages (C++ et Python dans notre cas). Le paradigme de ROS réside en 3 principes : les topics, les services et les actions. Les topics sont les canaux qui transmettent les données sous forme de messages. Les topics sont utilisés dans un but de communication continue et unidirectionnelle (canaux ou bus). Les interactions nécessitant des réponses sont prises en charge par les services. Les services sont définis par une paire de messages : un pour la question et un autre pour la réponse. La gestion des repères cartésiens se fait via les "tf". Les topics ROS se chargent de communiquer les systèmes de coordonnées locales et les transformées sous forme de messages à une fréquence souhaitée aux différents nœuds. Chacun des nœuds peut publier ou souscrire aux "tf" actuelles pour avoir une vision continue et globale du robot dans l'espace. ROS permet donc, de concentrer l'effort de développement sur les applications et tâches robotiques.



ArIA Inspector (Robot réel) ArIA Inspector (Simulation Gazebo)

FIGURE 4.7 – ArIA Inspector.

Le contrôle des servomoteurs est réalisé directement depuis l'ordinateur embarqué (Jetson Nvidia) grâce à l'U2D2 (échange d'informations en protocole UART). Dans cette

Catégorie	Caractéristiques	Valeurs
Vitesse de marche	Terrain régulier	0.1m/s
	Terrain accidenté	0.01m/s
Démarche	"Tripod, Amble, Ripple, Wave"	4 démarches
Posture	Garde au sol	Variable de 15 à 25 cm
Poids	Poids maximum	4.5kg
Capacité d'emport	Capacité d'emport	0.5kg
Obstacle avoidance	Hauteur d'obstacle	16cm
	Largeur d'obstacle	>6cm
Operation	Déploiement mécanique	Manuel
	Communication	Wi-Fi 2.4 GHz (& Radio Pixhawk)
	Capacité Omnidirectionnel	6 mouvements
	Autonomie	> 1h
Coût	Coût composant	<10K€
	Composants issue du commerce/sur mesure	Sur mesure : Batterie / Chassis Commerce : Capteurs/ Electronique

FIGURE 4.8 – Spécification de ArIA Inspector.

dernière version, un micro-contrôleur (Groveduino) sert à réaliser l'interface avec les capteurs de gaz. D'autres capteurs peuvent être intégrés comme les capteurs de pression, d'hydrométrie, etc. Les données peuvent ainsi être collectées et dirigées vers les différents nœuds ROS grâce aux topics permettant la communication. Le modèle 3D est directement implémenté à l'aide d'un fichier URDF [218], assemblant les différentes parties du robot et les capteurs. Ce modèle est prévu pour l'expérimentation et l'amélioration des futures versions.

L'architecture matérielle d'ArIA Inspector est décrite Figure 4.9. ArIA Inspector reprend les codes des architectures embarquées en utilisant un ordinateur embarqué avec GPU (Jetson Nvidia) et plusieurs capteurs comme le laser (RPLidar). Le laser est utilisé pour l'odométrie et le procédé de cartographie. Les capteurs visuels comme une caméra stéréoscopique (RealSense D435i) donnent une information de nuages de points où l'on peut extrapoler une mesure de profondeur (distance Z). De plus, cette caméra possède aussi une centrale inertielle (IMU intégré). Une carte Arduino munie d'un Grove Shield permet de réaliser des mesures sur des capteurs différents, ici des capteurs de gaz MQ3 (MOX). Une Pixhawk est aussi placée afin d'obtenir une mesure de l'assiette du robot. Les hubs TTL sont directement connectés aux servomoteurs (côté gauche et côté droit), un pont TTL les relie pour assurer le transfert en sortie sur l'U2D2 qui sert à communiquer les informations des servomoteurs (commandes d'angles, erreurs, couple, etc.) en USB et

est ensuite interprété sous ROS par le paquet ROS Dynamixel.

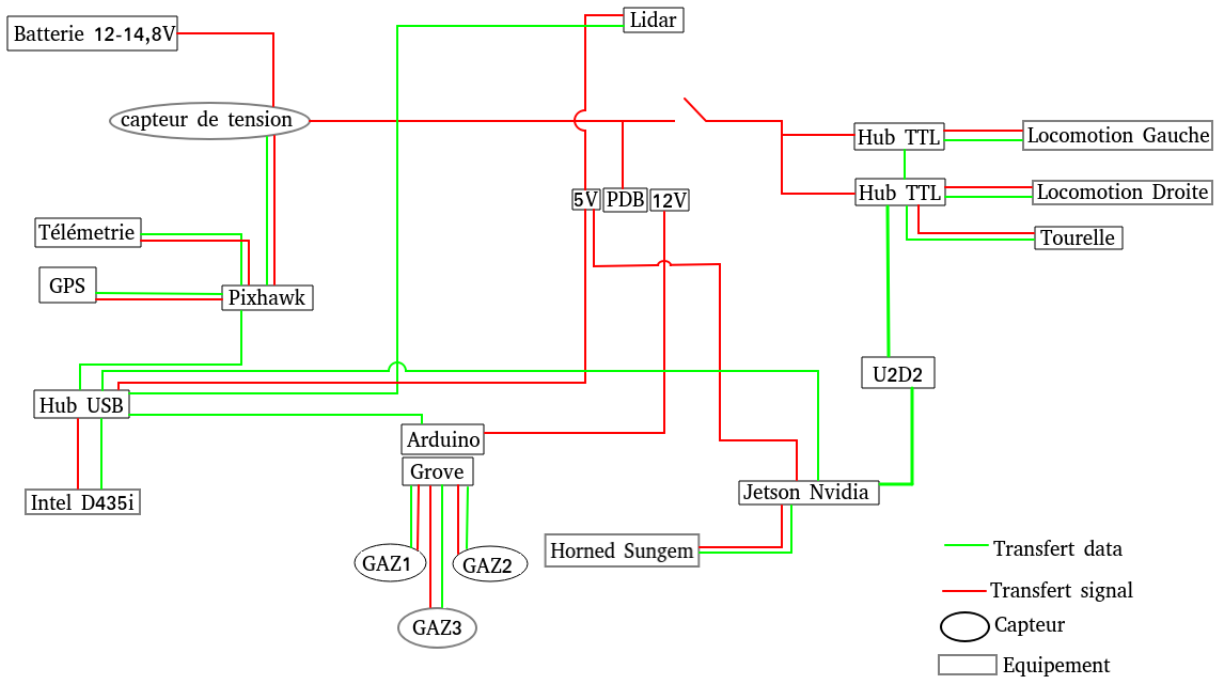


FIGURE 4.9 – Architecture de ArIA INSPECTOR.

4.2 Architecture du contrôleur

Nous avons évoqué la problématique du contrôle d'un robot marcheur tel qu'un robot hexapode. Son contrôle est basé sur plusieurs étapes : le contrôleur de pose, le contrôleur de marche et le contrôleur du robot comme illustré sur la Figure 4.10.

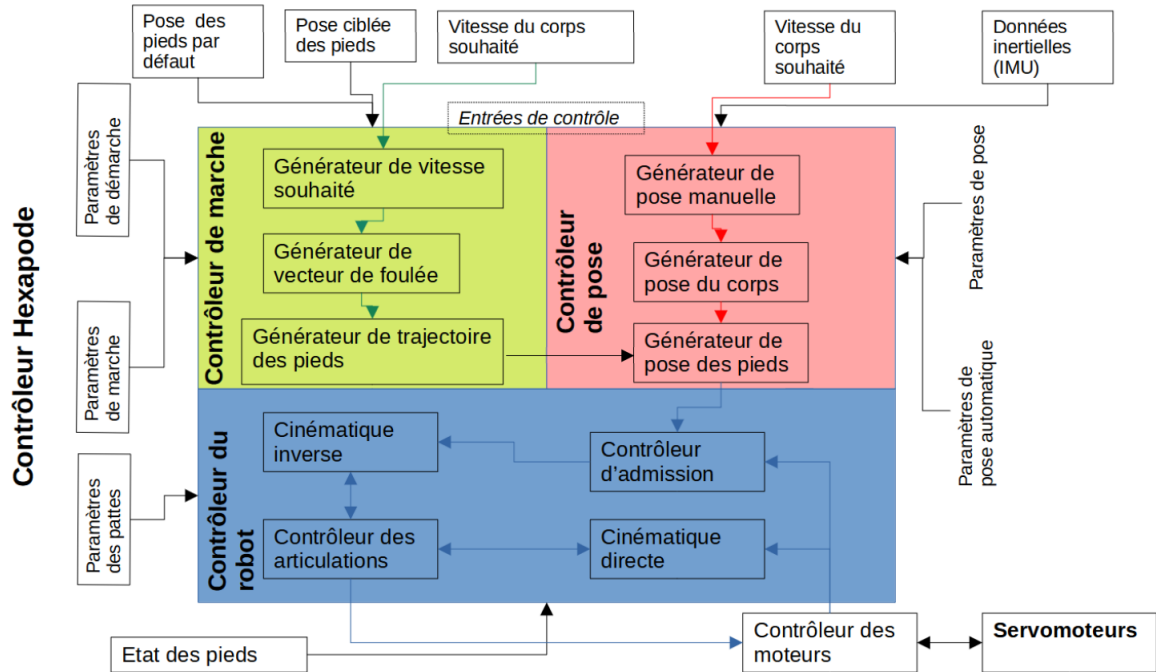


FIGURE 4.10 – Approche détaillé du contrôleur de l'hexapode OpenSHC [211]

Nous avons opté pour le contrôleur OpenSHC pour le contrôle de l'hexapode et la librairie ROS "move_base" comme infrastructure de navigation. OpenSHC [211] reçoit les vitesses souhaitées (avant, latérales et angulaire) et les vitesses de pose du corps (6 DOF) via une source externe. La source peut être un opérateur via une manette de jeu (manette Xbox360), tablette, ordinateur, ou depuis un procédé de navigation autonome. L'interface prend un vecteur vitesse linéaire sur les axes x , y , z et un vecteur de vitesse angulaire autour des axes x , y , z , en utilisant les messages ROS `geometry_msgs::Twist`. Pour le contrôleur de marche, le système utilise les vitesses linéaires x et y et vitesse angulaire par rapport à l'axe z . La vitesse linéaire sur l'axe z et les vitesses angulaires autour de l'axe x et y sont ignorées. Pour le contrôleur de pose, un vecteur de vitesse linéaire et angulaire est utilisé pour poser le corps du robot. L'utilisateur peut sélectionner la source externe à utiliser, permettant la navigation automatique ou le mode régulateur de vitesse, ainsi que le contrôle manuel. La navigation automatique nécessite des commandes de la

navigation autonome qui permet d'éviter les obstacles. Le régulateur de vitesse définit la vitesse constante du robot dans le générateur de vitesse souhaité dans la démarche.

Le contrôleur de l'hexapode est implémentable sur ROS et Gazebo. Ce contrôleur nous permet de prendre en compte la géométrie du robot, les moteurs, la taille des segments pour calculer les fins d'effecteurs, l'espace de travail, l'espace de marche. Autrement dit, les différentes zones de liberté de la fin d'effecteur par rapport aux paramètres intrinsèques du robot. Le contrôleur utilise les courbes de Bézier pour générer les trajectoires des pattes. L'odométrie est calculée par le contrôleur, mais reste approximative et un capteur extéroceptif ou proprioceptif (ou les deux) est nécessaire afin d'assurer un minimum de robustesse dans le procédé de navigation.

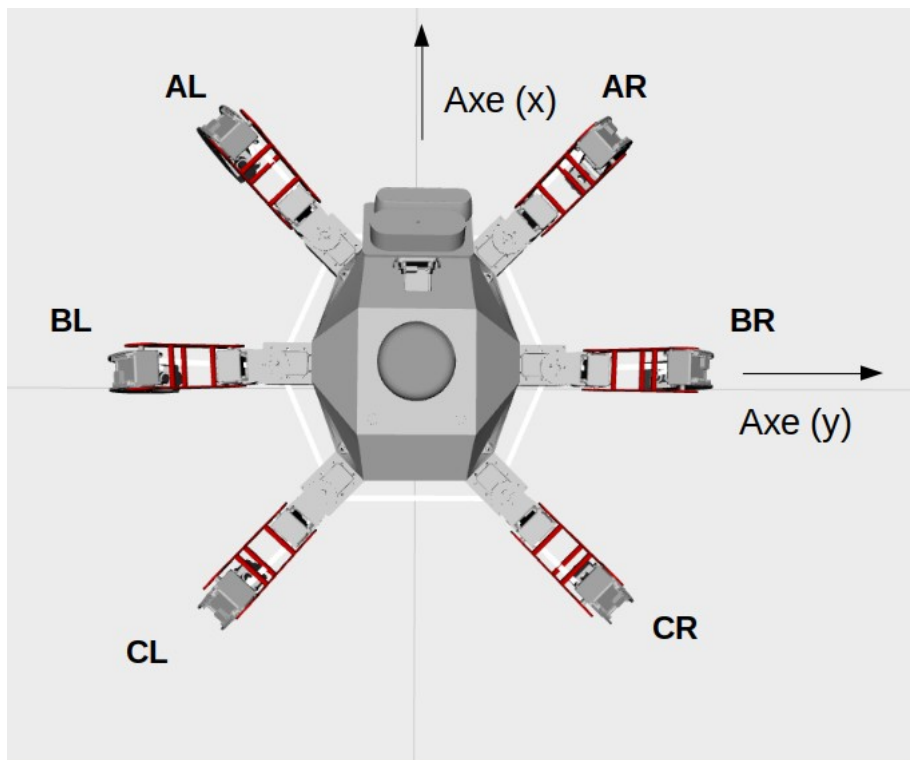


FIGURE 4.11 – Disposition des pattes d'ArIA Inspector

Les différents segments des pattes sont notés à partir d'un couple de lettres, Figure 4.11. Pour faciliter la lecture, l'emplacement des pattes est défini par la première lettre et le côté des pattes par la deuxième lettre. Les pattes avant sont notées par la lettre A, celle du milieu par la lettre B et enfin les pattes arrière par C. Le côté gauche par la lettre L ("Left") et le côté droit la lettre R ("Right"). Les paramètres du modèle (Matrices

Denavit-Hartenberg) sont intégrés directement dans un fichier de configuration `.yaml`. À partir du fichier de configuration, le modèle peut être généré avec son espace de travail et son espace de marche (Figure 4.12). L'espace de travail est caractérisé par la morphologie et la limite des joints du modèle concerné.

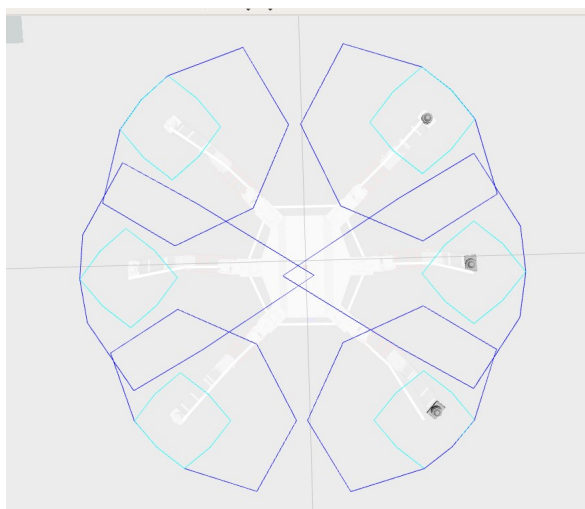


FIGURE 4.12 – Espace de travail (bleu foncé) et Espace de marche (bleu) de ArIA Inspector.

Le contrôleur OpenSHC [211] se caractérise par 3 niveaux. La pose, la démarche et le robot (Figure 4.10). La géométrie du robot est utilisée en données d'entrées pour la cinématique inverse (Chapitre 3) et différents paramètres peuvent s'appliquer aux effecteurs (raideurs, amortissements, etc.). La cinématique permet d'avoir la pose de l'hexapode et ainsi d'avoir la position des fins d'effecteurs. À partir de là, les positions désirées et les positions manuelles sont possibles. Grâce à ses positions de fin d'effecteurs, la démarche peut être réalisée en cadencant les oscillations des pattes et on peut obtenir plusieurs démarches. La trajectoire des fins d'effecteurs est réalisée par les courbes de Bézières. Afin d'utiliser le contrôleur en simulation, il sera aussi nécessaire de réaliser les réglages de réponse de signal pour les servomoteurs : les réglages PID.

4.2.1 Méthode de Réglage PID : Simulation

Des réglages PID (Proportionnelle Intégrale Dérivée) du signal de commande des moteurs dans la simulation sont nécessaires. Ces réglages permettent d'obtenir une réponse et une amplitude de signal adaptée entre la commande et le signal de réponse des effecteurs. Les réglages PID sont réalisés en utilisant le paquet ROS PID. Les paramètres sont générés dans un fichier de configuration `Pid.cfg` voir Annexe.

L'autoréglage des PID est réalisé avec la méthode Ziegler-Nichols [243]. Cette approche propose de calculer les paramètres du régulateur P, PI ou PID à l'aide des recommandations suivantes pour le régulateur :

Etape 1 de régulation : Proportionnelle (K_p).

$$P : R(p) = K_p \quad (4.1)$$

Avec T_i et $T_d = 0$.

Etape 2 de régulation : Proportionnelle (K_p) Intégrale (T_i)

$$PI : R(p) = K_p \left(1 + \frac{1}{T_i p}\right) \quad (4.2)$$

Avec $K_p = \frac{T_u}{K} \frac{0.9}{T_u}$ et $T_i = 3.33T_u$

Etape 3 de régulation : Proportionnelle (K_p) Intégrale (T_i) Dérivée (T_d).

$$PID : R(p) = K_p \left(1 + T_d p + \frac{1}{T_i p}\right) \quad (4.3)$$

Avec $K_p = \frac{T_u}{K} \frac{1.2}{T_u}$, $T_i = 2.0T_u$ et $T_d = 0.5T_u$.

4.3 Architecture de navigation

Nous procédons à la cartographie à l'aide d'un laser Lidar pour réaliser l'environnement de cartographie afin de traiter l'asservissement visuel. Pour l'application d'asservissement chimique, la cartographie est déjà traitée et un fichier de carte où le CFD est utilisé avant les opérations. La navigation (ROS) a différentes fonctionnalités : déplacer la base avec la carte de coût et le planificateur de trajectoire, mais aussi un comportement de récupération, voir Figure 4.13. Le planificateur de navigation utilise le nœud de base de

déplacement. Il permet d'appliquer différents paramètres en fonction de la géométrie et du types de robots. Nous avons choisi l'algorithme AMCL comme méthode de localisation en raison de la bonne combinaison avec une méthode de mise en correspondance par balayage laser (Laser Scan Matcher). L'odométrie à partir de l'empreinte de base n'est pas fiable, car il peut s'agir d'un bruit de mouvement (jambe glissante, décalage mécanique, etc.). Des capteurs extéroceptifs sont nécessaires pour améliorer la fiabilité de la navigation. AMCL permet d'estimer la pose du robot à l'aide de la carte et de l'empreinte de base. La fonction de pose de but sur Rviz permet de déplacer la base (sujets de vitesse du robot) vers le robot sur Gazebo (cycle des jambes, démarche). La configuration de la locomotion peut avoir un impact sur la fiabilité de la navigation en raison de la complexité du mouvement et du bruit impliqués.

4.3.1 Implémentation de la navigation

La navigation est basée sur l'obtention de la carte de l'environnement par le capteur lidar comme décrit Figure 4.13.

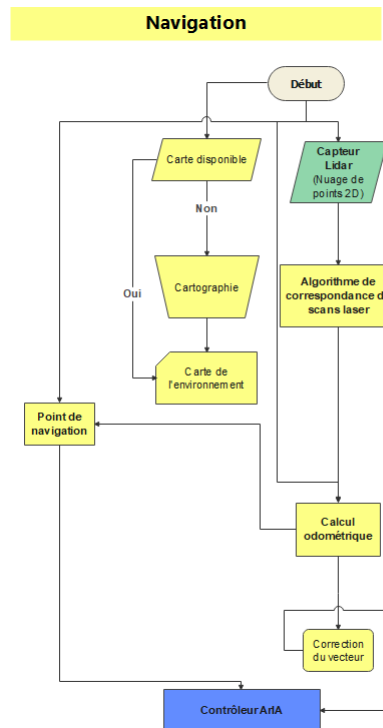


FIGURE 4.13 – Logigramme de navigation.

Afin de décrire plus précisément la partie navigation, nous illustrons, Figure 4.14, le

diagramme de la navigation. Le procédé de navigation sera composé de la source odométrique, en l'occurrence donnée par la méthode de correspondance de scans laser en utilisant la différence de captation entre 2 scans du capteur lidar 2D. Par ailleurs, d'autres informations peuvent enrichir les données odométriques, comme les données des capteurs IMU contenu dans la caméra stéréoscopique.

Dans le diagramme de navigation, nous utilisons différents outils que constitue la brique logicielle "Move Base" :

- Le "local costmap" : concerne les paramètres d'envergure liés au robot.
- Le "global costmap" : concerne les paramètres englobant les contours de la carte (dans l'espace naviguable), ajoute une inflation sur celle ci.
- Le "local planner" : réalise les fonctions de coût liées aux obstacles.
- Le "global planner" : réalise la planification de trajectoire, dans notre cas, nous utiliserons l'algorithme Dijkstra [199].

L'avantage de ce système est de pouvoir paramétrer facilement les réglages de navigation influant sur le comportement du robot dans le ralliement d'un point de navigation.

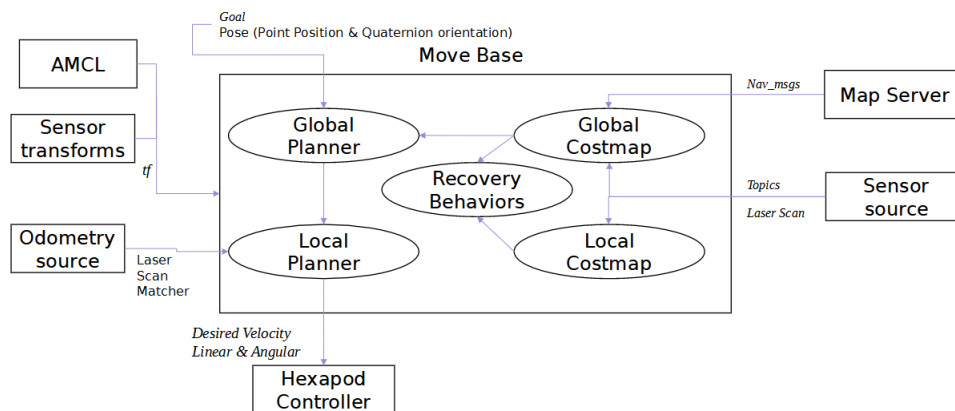


FIGURE 4.14 – Méthode de navigation en utilisant le paradigme de ROS.

L'odométrie est la capacité d'un robot à estimer sa localisation et sa pose. La difficulté d'estimer la pose d'un robot hexapode vient d'une succession de mouvements de jambes qui permettent le mouvement de base. Différente du robot à roues, l'odométrie est calculée via une vitesse souhaitée basée sur la vitesse du corps, mentionnée précédemment dans la section sur le modèle cinématique. L'architecture de navigation est décrite Figure 4.14. Afin de réaliser un procédé de navigation fiable, nous mettons en place la méthode de la correspondance de scans laser comme source odométrique et cartographique en 2

dimensions. La vitesse linéaire et angulaire provenant du contrôleur est utilisée pour déplacer le robot de manière autonome. La carte des coûts prend en compte les données des capteurs avec un repère global lié à la carte. Les coûts sont augmentés en fonction d'une grille d'occupation et d'un rayon d'inflation correspondant aux paramètres de locomotion et géométriques du robot. Le rayon d'inflation est un paramètre de coût associé à l'environnement et au robot, qui permet de définir une zone de sécurité pour éviter la collision entre le robot et l'environnement. La grille d'occupation permet d'éviter les zones létales en ajoutant une zone de sécurité. Le planificateur organise les trajectoires du robot. L'algorithme Dijkstra est utilisé pour la planification des trajectoires.

Pour ce robot, nous choisissons 4 types d'odométrie : l'estimation de pose EKF qui traite la fusion et le filtrage de toutes les données d'odométrie en sortie, l'odométrie (à partir des résultats de vitesse souhaités), la correspondance de scans laser met à jour la localisation du robot. Il utilise les différences entre deux trames de sortie laser. Concernant le paquet logiciel de navigation, `move_base` est utilisé pour réaliser l'architecture de navigation. Il est fourni nativement par ROS et permet de configurer les différents paramètres de navigation comme les fonctions de coûts utilisées pour le déplacement du robot par rapport à son environnement, l'emplacement du robot sur la carte ou encore la gestion des trajectoires. Nous avons présenté ici l'intégration de l'architecture du robot et son intégration logicielle. Les briques présentées constituent l'architecture de base permettant de s'appuyer sur les méthodes d'asservissements utilisées dans cette étude.

4.4 Asservissement visuel

L'asservissement visuel permet au robot de se diriger de manière automatique vers un percept visuel via une boucle de commande. Le thème de l'asservissement visuel est un sujet qui peut conduire à exploiter pleinement les capacités motrices et géométriques d'un robot hexapode. L'objectif de cette intégration est de prouver que le robot est capable de se déplacer en fonction de ce qu'il a dans son champ de vision. Pour cela, un paquet ROS est utilisé qui permet au robot de suivre une cible en mouvement avec un marqueur aruco, Figure 4.15. La caméra est utilisée pour détecter le QRcode. Le robot peut ainsi calculer la transformation géométrique entre ce marqueur et lui-même et publier des destinations de navigation basées sur celui-ci via le nœud de base de déplacement. Les transformées sont alors extraites et un jacobien d'image sur un point est réalisé. Le robot se dirige dans une direction sur un plan 2D. Un service de suivi de marqueur est appliqué pour diriger automatiquement le robot même si la cible devient dynamique ou si elle est perdue de vue. Le procédé de navigation et d'évitements d'obstacles est appliqués avec les méthodes vues précédemment. L'approche visuelle est séparée en plusieurs parties. Au cours de la première partie, nous allons décrire une approche par image de référence (QR code), afin de démontrer le procédé en simulation. La seconde partie évoque l'approche par réseaux de neurones convolutifs.

4.4.1 Application de l'asservissement visuel par le QR Code

Dans cette approche, (Figure 4.15), nous utilisons le flux visuel de la caméra comme donnée d'entrée pour détecter le QR Code via l'algorithme "fiducial" pour récupérer un repère cartésien dans le repère monde. Ce repère identifié, le robot va donc établir une trajectoire en prenant le repère QR Code comme point de navigation.

Cette approche est entièrement intégrée dans l'architecture ROS. Sur la figure 4.15, nous présentons l'architecture logicielle. Les services `/spawn_urdf`, `/gazebo_gui` et `/map_server` sont les services associés respectivement au modèle 3D du robot, au service de simulation et à la carte de l'environnement. Les services `/aruco_detect` et `/fiducial_images` sont les services pour détecter le QR Code et produire le repère cartésien (du QR Code) qui servira comme point de navigation. En simulation, nous avons attaché un QR Code à un robot roulant (Turtlebot) qui servira de cible. Cette approche nous permet, à partir d'une image de référence, d'asservir le déplacement du robot vers le percept visuel. Les commandes du service `/fiducial_images` sont ainsi envoyées dans le service `/move_base`

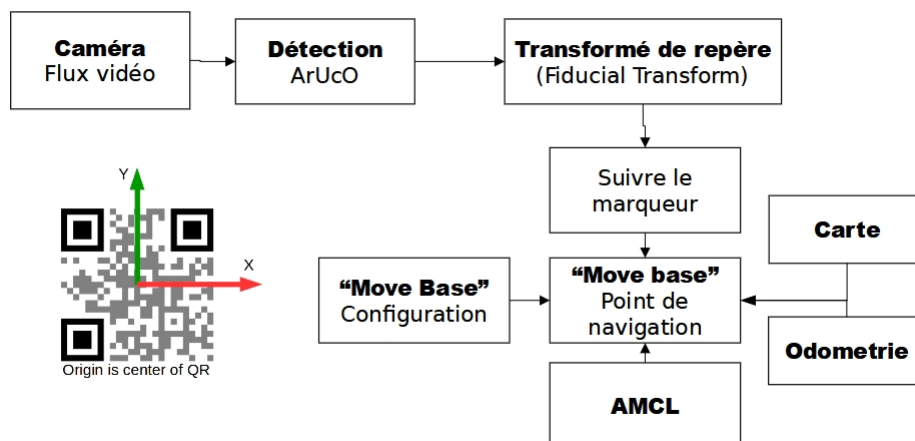


FIGURE 4.15 – Asservissement visuel en utilisant un QRcode.

qui produira le point de navigation et enverra les commandes au robot par le service `/cmd_vel` sous la forme d'une vitesse linéaire et angulaire.

4.4.2 Application de l'asservissement visuel par la reconnaissance d'objets

L'architecture de la reconnaissance et l'asservissement visuel est décrite sur la Figure 4.16. Nous utilisons l'architecture YOLO v5 permettant l'application d'une boîte englobante sur l'objet détecté.

La librairie PyTorch est utilisé pour générer le réseau convolutif. Le détail de l'architecture YOLO v5 est présentée sur la figure 4.17, où les différentes couches de convolution et de concaténation sont décrites. L'architecture de YOLO v5 (S) est composée de 3 parties : l'ossature du modèle ("Model Backbone"), le cou du modèle ("Model Neck") et la tête du modèle ("Model Head"). L'ossature du modèle est principalement utilisée pour extraire les informations caractéristiques des images d'entrée grâce aux réseaux partiels à plusieurs étapes ("CSP - Cross Stage Partial Networks"). Le cou du modèle est utilisé pour créer une pyramide de caractéristiques, servant à mieux généraliser un même objet possédant des tailles ou couleurs différentes. La tête du modèle est utilisée pour détecter et identifier l'objet dans l'image.

La base de données est constituée de fissures et de différents défauts industriels (défauts de soudures, rayures sur le métal, etc.). Cette base de données a été augmentée une première fois, avec des filtres différents de couleurs (rouge, bleu, vert). Puis la base est augmentée une deuxième fois, avec des filtres de bruit d'images, de différences de contraste

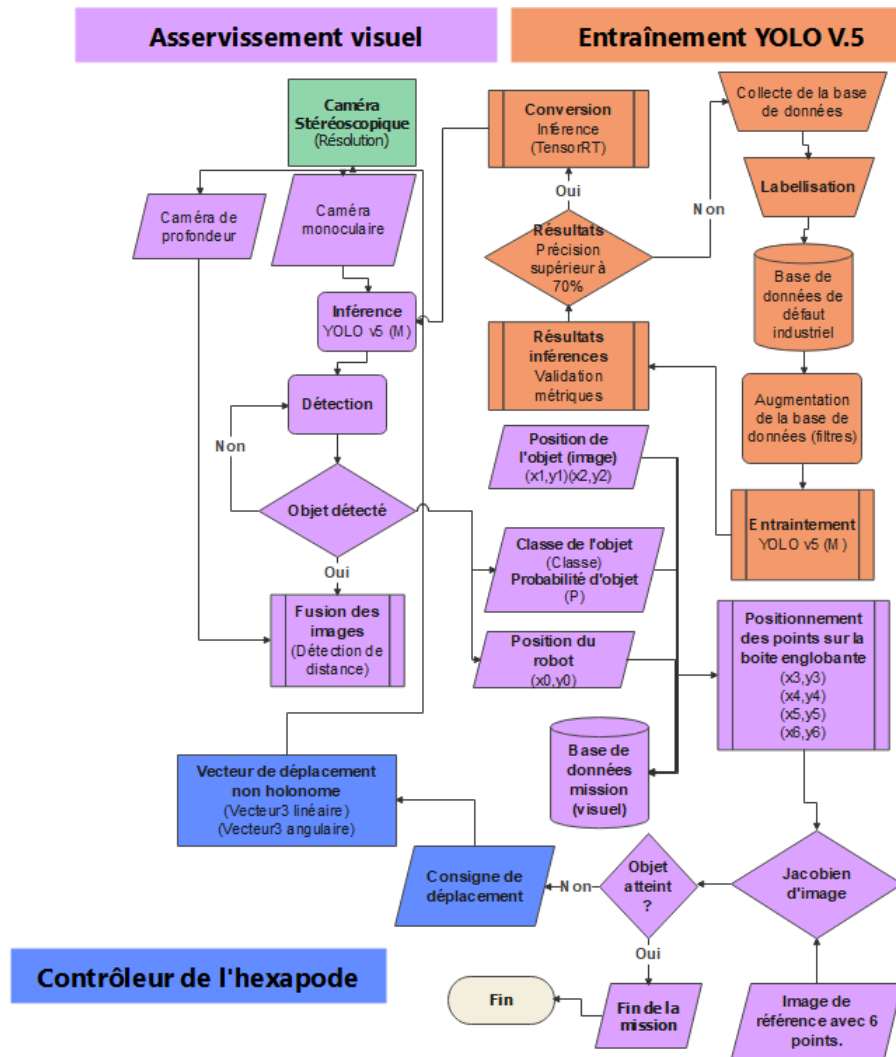


FIGURE 4.16 – Logigramme de l’asservissement visuel par reconnaissance d’objets

et de luminosité. La base de données produite contient 2700 images de fissures, et environ 500 images de chaque catégorie de défauts (crazing, inclusion, patches, pitted surface, rolled in scale, scratches). Cette base de données est ensuite envoyée dans le réseau de neurones pour l’entraînement.

Le jacobien visuel, présenté dans le chapitre 3, permettra de faire le lien entre l’emplacement de l’objet dans l’image et la direction à définir pour le robot.

Nous avons ici présenté deux méthodes d’asservissement visuel. La méthode par QR code est intégrée dans une architecture ROS. La problématique majeure d’un procédé d’asservissement visuel est l’erreur qui peut se cumuler par différents facteurs : les erreurs de localisation, mécanique, visuelle et de navigation. Dans les procédés d’essais, nous

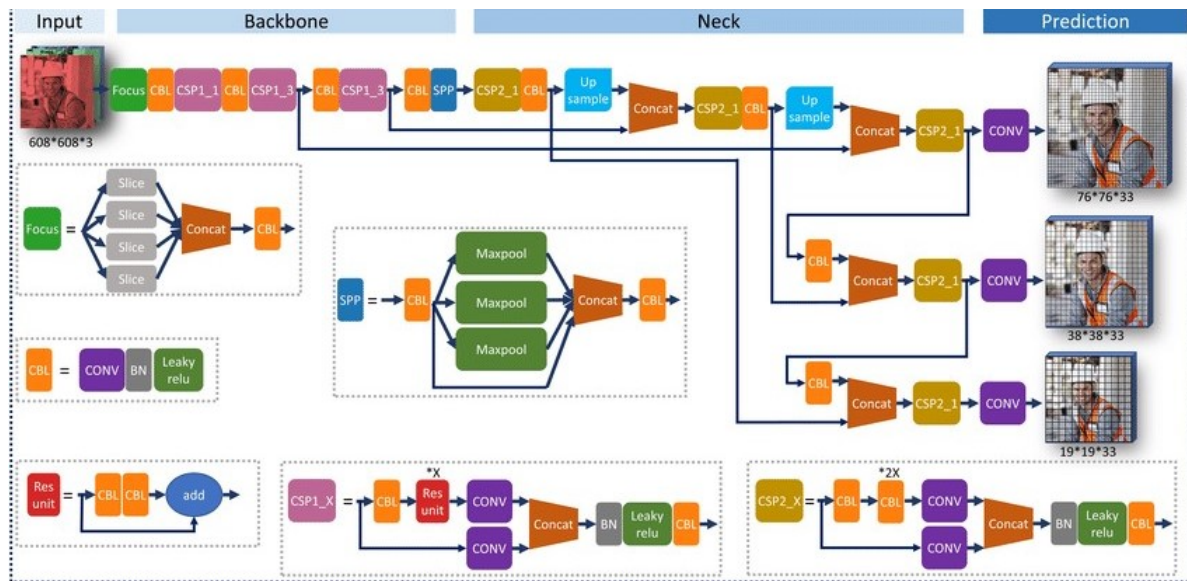


FIGURE 4.17 – Architecture des différentes couches de YOLO V5 (S) [242]

avons choisi de tester les performances de navigation, de localisation et d'odométrie, afin de réduire au maximum l'erreur cumulée sur les différentes briques technologiques utilisées.

4.5 Asservissement chimique

Dans cette section, nous allons aborder l'intégration de l'approche d'asservissement chimique par les capteurs, l'environnement de simulation et l'algorithme de recherche de gaz. La mise en œuvre de l'asservissement chimique a pour but de permettre au robot de planifier sa navigation en fonction de la présence d'une veine de gaz dans son environnement. Idéalement, le robot devrait se déplacer vers la source de gaz de manière autonome. Les gaz ciblés par les capteurs sélectionnés étant nocifs, la mise en place de l'expérience sur le vrai robot est prématurée. Pour pallier cette situation, le développement de l'asservissement chimique s'est déroulé en simulation [176].

4.5.1 L'approche chimique

Dans cette approche, (Figure 4.18), nous utilisons les capteurs de gaz et l'anémomètre comme donnée d'entrée pour générer des points de navigation via l'algorithme de recherche de gaz. Les points de navigation sont séquencés de manière itérative et en fonction des valeurs des capteurs.

Pour traiter l'asservissement chimique, l'algorithme de grille ("Grid") [208] est appliqué pour rechercher la localisation de la source de gaz de manière itérative pour atteindre le point de destination. Cet algorithme inclut la montée au vent dans le panache et le vent de travers lorsque le panache est perdu. Le capteur de gaz et l'anémomètre sont obligatoires en entrée. La direction au vent est calculée en fonction des données de l'anémomètre. Ensuite, les objectifs sont envoyés sur l'axe x, l'axe y et l'orientation sur l'axe z en fonction de la position de pose actuelle du robot sur l'axe x, l'axe y et l'orientation sur l'axe z. L'objectif est envoyé au nœud Move Base pour exécuter le chemin. L'algorithme est décrit par cinq états différents :

- 1) En attente de la carte : étape d'initialisation de la carte. Si la carte n'est pas disponible, une étape d'attente est configurée pour attendre que le serveur de carte en fournisse une.
- 2) État d'exploration : si la carte est disponible, alors elle recherchera les traces du dégagement gazeux.
- 3) État d'inspection : si un gaz est détecté, il recherchera dans les environs la source de gaz.
- 4) Panache (Grid) : si un panache de gaz peut exister, il essaiera de le diriger vers la source de gaz en envoyant des objectifs.
- 5) Etat d'arrêt et de mesure : une fonction de récupération et de mesure. Il arrête le robot

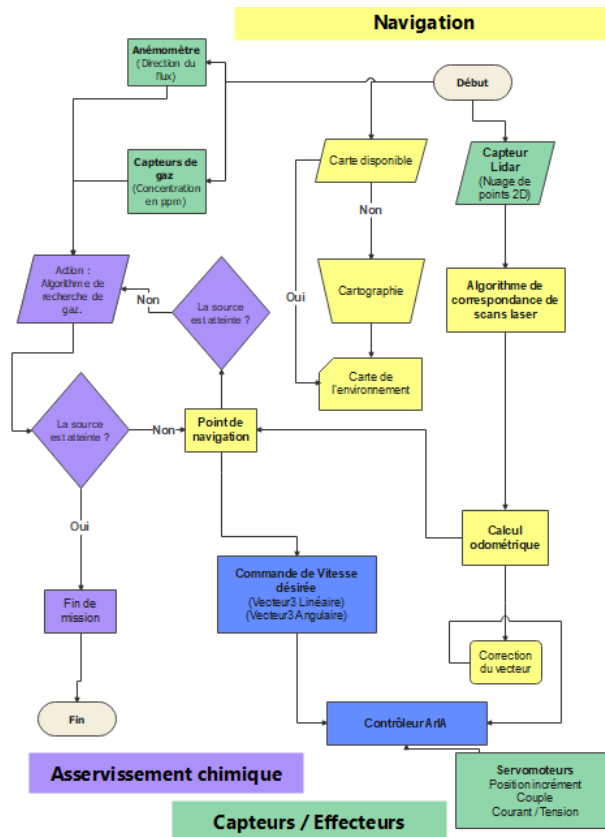


FIGURE 4.18 – Asservissement chimique en utilisant un algorithme de recherche de gaz.

et mesure le vent et le gaz pendant un laps de temps.

De récentes recherches [164] et [176] ont été menées sur le développement d'un simulateur de dispersion de gaz avec une intégration possible sur ROS. Ce simulateur permet de simuler des capteurs de configurations différentes.

4.5.2 L'environnement de simulation

Sur la figure 4.19 est présenté le fonctionnement de l'environnement de simulation de fuite de gaz GADEN. La première partie est le pré traitement, le modèle de l'environnement 3D (en format mesh, 3D) est utilisé pour calculer les flux dynamiques et générer une base de données de flux d'air. Le simulateur va alors générer le panache de gaz en fonction des données du flux d'air calculé pour l'environnement et les paramètres des sources de gaz. La dispersion du gaz est alors recrée dans l'environnement de simulation en boucle. Les capteurs sont aussi simulés dans l'environnement et permettent d'opérer des essais sur l'acquisition avec différents facteurs de bruits capteurs possibles.

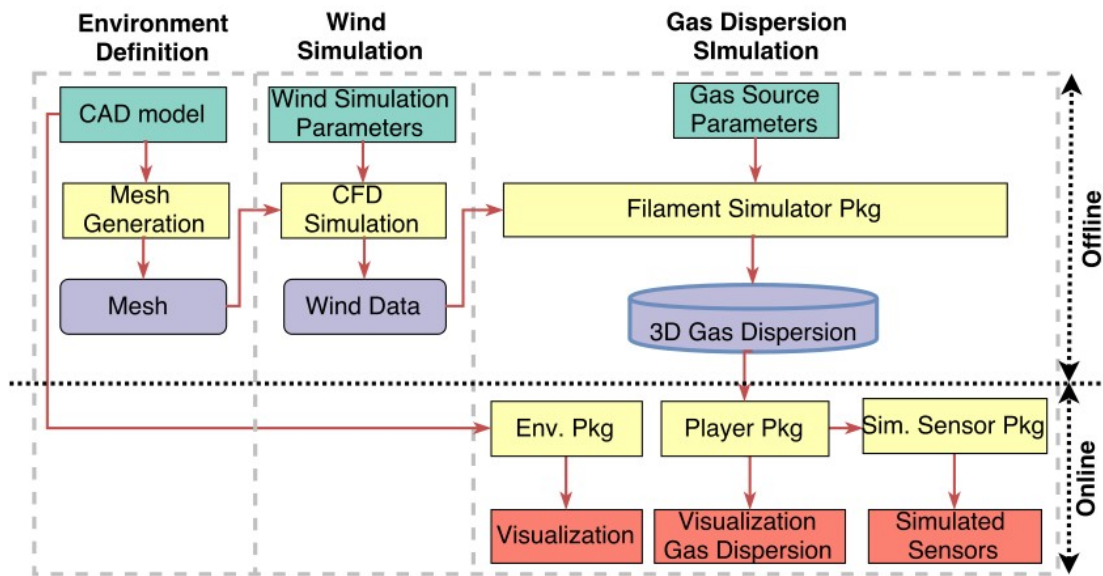


FIGURE 4.19 – Architecture de traitement de l'environnement de simulation de gaz [176]

Pour les essais, nous avons utilisé des environnements fournis par le paquet GADEN, mais il est tout à fait possible de générer le flux d'air d'un environnement à partir d'un environnement personnalisé à partir de son modèle 3D et d'une interface de simulation CFD.

4.6 Conclusion de fin de chapitre

Dans ce chapitre, nous avons évoqué l'intégration des algorithmes d'asservissement visuel et chimique. Afin d'appréhender les différents aspects de la robotique d'un point de vue logiciel, nous utilisons ROS comme architecture logicielle et Gazebo comme moteur de simulation. Le premier challenge est d'avoir des données odométriques suffisamment fiables pour localiser le robot hexapode. Le deuxième sujet est de pouvoir déplacer le robot dans des lieux contraints avec des ouvertures faibles. Le troisième challenge est de pouvoir réaliser un asservissement visuel avec un robot marcheur, en sachant que sa locomotion peut entraîner des perturbations (vibrations sur l'image) et altérer la détection du percept. La quatrième problématique est de pouvoir réaliser un asservissement chimique dans des environnements contraints avec une gestion du planificateur de trajectoire adapté à ce type d'environnement. Les approches déployées vont être testées en simulation en les comparant aux positionnements de la vérité terrain (positionnement absolu) du robot et des différents percepts.

EXPÉRIMENTATIONS ET RÉSULTATS

Comme nous l'avons évoqué dans les chapitres précédents, le but de cette thèse est de mettre en place une méthodologie pour la détection automatique de défauts dans un milieu contraint. Nous avons proposé, une approche d'asservissement visuel et chimique afin de réaliser de l'inspection industrielle automatisée à l'aide d'un robot hexapode ArIA Inspector. Dans ce chapitre, la première partie décrit les expérimentations en simulation et résultats de l'odométrie, la navigation, ainsi que les procédés d'asservissement. Dans la seconde partie, nous présentons les expérimentations et résultats d'analyse d'efforts statiques et dynamiques sur le robot réel. Enfin, dans la troisième partie, nous présentons les expérimentations menés sur des scénarios représentatifs, une simulation dans un labyrinthe et un essai en situation réelle dans une course de bateau.

5.0.1 Introduction

La cohabitation entre les paramètres de navigation et le procédé d'asservissement chimique ou visuel d'un robot hexapode est assez complexe, car le robot développé est destiné à opérer dans un environnement contraint et dégradé. Les essais de la plateforme développée ont pour but de déterminer les performances et le comportement du robot dans une mission d'asservissement. Les méthodes d'asservissement seront réalisées par le robot d'expérimentation (ArIA INSPECTOR) et dans un environnement de simulation Gazebo.

5.1 Expérimentations en simulation

Dans cette section, nous allons aborder le protocole expérimental pour les essais en simulation : l'odométrie, la navigation et l'asservissement.

5.1.1 Protocole expérimentale pour la simulation

Nous présentons dans la première partie les résultats de l'odométrie, la navigation, l'asservissement visuel et chimique concernant le robot simulé. Puis, dans la seconde partie, nous abordons les résultats des efforts statiques, dynamiques et de franchissement du robot hexapode en situation réelle. La synthèse des essais est présentée Figure 5.1. Les essais odométriques vont déterminer la source odométrique la plus fiable parmi l'IMU, la vitesse désirée et calculée par le contrôleur OpenSHC, la fusion de capteurs (EKF) et la méthode de correspondance laser. Pour tester les capteurs, nous appliquons un mouvement programmé : une trajectoire en carré de 2 mètres sur 2.

La navigation (Figure 5.1) est testée sur un point en milieu contraint avec des passages de 1 et 1,5 mètre. Le but est de déterminer si les réglages et algorithmes de navigation sont suffisants pour atteindre ce type d'environnement et en dégagé une potentielle erreur de localisation ou de navigation durant le mouvement.

L'asservissement visuel (Figure 5.1) est testé sur une cible fixe en utilisant un robot (Turtlebot3) qui tiendra la cible de manière statique. Le but est de vérifier l'erreur de positionnement du robot hexapode par rapport à sa cible.

L'asservissement chimique (Figure 5.1) est décomposé en plusieurs parties : la partie de prétraitement va décomposer le flux d'air pour qu'il soit traité en positionnant la source du gaz. La visualisation du gaz permet alors de voir les particules de gaz et le panache. L'asservissement chimique débute une fois que l'environnement permet la visualisation des particules. Les essais d'asservissement chimique dans le milieu A (environnement avec un obstacle) et le milieu B (labyrinthe) permettront de qualifier la vitesse d'asservissement chimique, l'erreur de positionnement et d'orientation.

Les expérimentations ont pour but de vérifier la robustesse de l'odométrie, le procédé de navigation, l'asservissement visuel et l'asservissement chimique par différents indicateurs de performance (l'erreur, la position, la durée d'asservissement, etc.). L'expérimentation est réalisée sur Gazebo comme moteur de simulation et RViz comme outils de visualisation 3D pour l'odométrie, la navigation, l'asservissement visuel et l'asservissement chimique. La synthèse (Figure 5.1) montre les différents essais réalisés.

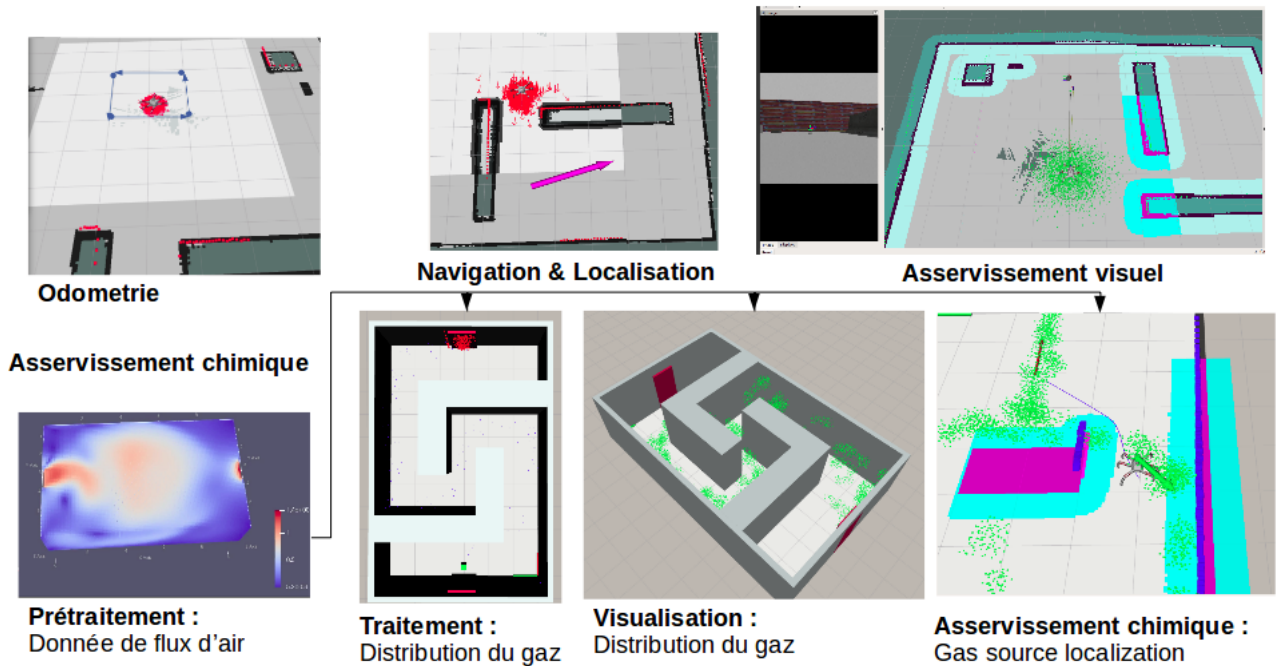


FIGURE 5.1 – Illustration des différents essais opérés sous ROS et Gazebo sur l'odométrie, la navigation et la localisation, l'asservissement visuel et l'asservissement chimique.

5.1.2 Expérimentation et résultats de l'odométrie

Le protocole de l'expérimentation et les résultats de l'odométrie sont présentés.

Protocole d'expérimentation de l'odométrie

L'odométrie est testée en réalisant une loi de commande pour une trajectoire du robot : un carré de 2 mètres sur 2 mètres (figure 5.2). Les publications des `topics` sont composées de la fusion de capteurs par la méthode d'un filtre de Kalman étendu ("EKF pose estimation"), de l'odométrie (calculé par le contrôleur), la méthode de correspondance laser ("Scan Matcher" [210]) et la vérité terrain (position absolue dans la simulation Gazebo) sont utilisées sur le même cadre de référence, le corps principal de l'hexapode. Les essais odométriques vont déterminer si la source odométrique la plus fiable parmi l'IMU, la vitesse désirée et calculée par le contrôleur OpenSHC, la fusion de capteurs (EKF) et la méthode de correspondance laser. Pour tester les capteurs, nous appliquons un mouvement programmé : une trajectoire en carré de 2 mètres sur 2. En utilisant ROS, il est possible de créer des fichiers regroupant l'ensemble des `topics` publiés (l'odométrie, la

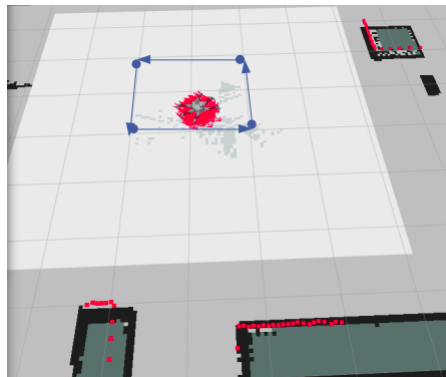


FIGURE 5.2 – Le test d’odométrie avec le robot ArIA Inspector.

vitesse linéaire et angulaire du robot, son positionnement, etc.). Les données sont enregistrées dans des fichiers de type `rosvbag` avec l’ensemble des `topics` publiés, puis segmenté en classeur au format `csv`. L’enjeu est d’obtenir des informations de capteurs suffisamment fiables pour avoir une loi de commande correcte. De plus, le positionnement du robot dans son espace joue aussi un rôle majeur dans le cadre d’un asservissement avec un système de navigation.

Résultats de l’odométrie

Les sources sont mises sous forme de graphique. Dans ce procédé, nous testons différents algorithmes afin de déterminer le plus adapté pour un robot marchant

Pendant le test d’odométrie, les données EKF de l’estimation de la pose (à l’aide de l’IMU) et l’odométrie du contrôleur (vitesse souhaitée) dérivent rapidement en raison de l’erreur accumulée. La correspondance par balayage laser est la méthode la plus appropriée pour obtenir une odométrie précise par rapport aux données de vérité terrain, voir figure (5.3).

À partir de la figure (5.4), nous pouvons voir une odométrie bruitée de l’IMU par rapport à celle calculée à l’aide du contrôleur. Ce problème est souvent lié à un paramètre de bruit simulé, reproduisant les conditions réelles. Nous avons finalement utilisé la méthode de correspondance de balayage laser [210] pour traiter la localisation du robot hexapode.

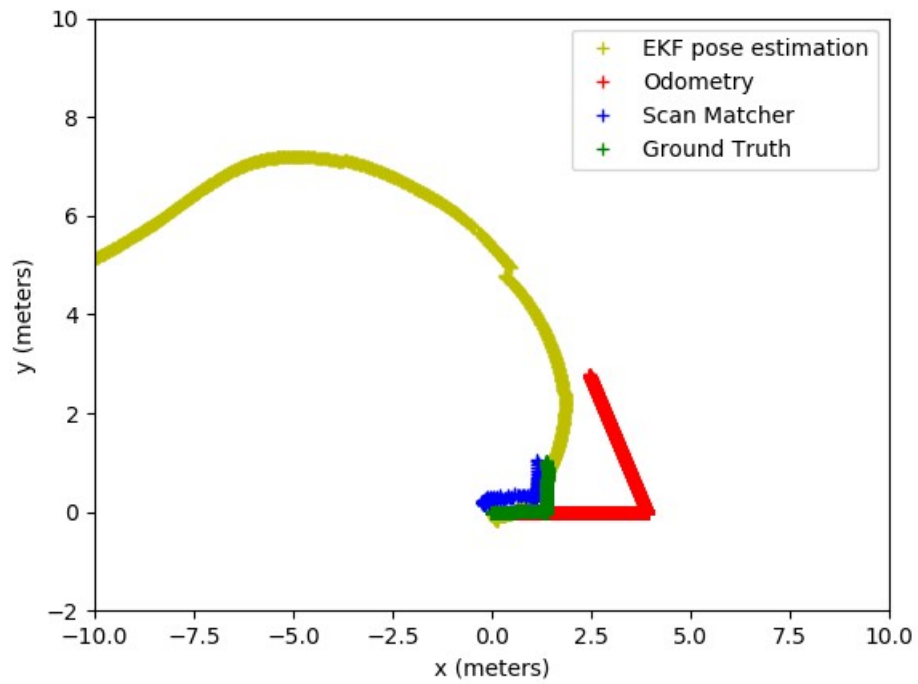


FIGURE 5.3 – Comparaison d'odométrie à partir d'une trajectoire.

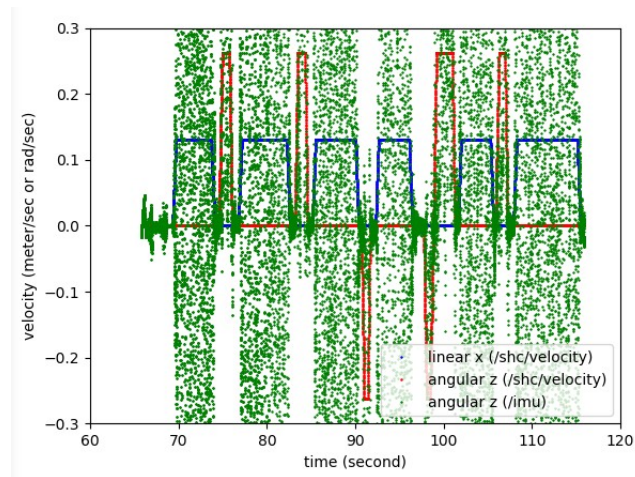


FIGURE 5.4 – Données IMU comparées aux données de vitesse souhaitées.

5.1.3 Expérimentation et résultat de la navigation

Expérimentation de la navigation

Un procédé de navigation recueille les informations des capteurs et de l'odométrie pour émettre des commandes de vitesse à envoyer à une base mobile. L'enjeu du problème est de diriger un robot marcheur (hexapode) vers un point de passage dans un environnement contraint. La navigation est testée en envoyant le robot dans un espace de 4 mètres sur 4, Figure 5.5, avec une entrée principale de 1 mètre. L'empreinte du robot est basée sur le rayon atteignant les fins d'effecteurs. Ce rayon est défini comme la distance euclidienne entre les fins d'effecteurs et le centre du corps. Le champ de particules représente des localisations possibles générées par la méthode AMCL ([194]), en rouge, Figure 5.5. L'essai permettra de déterminer l'erreur de localisation, qualifier la trajectoire.

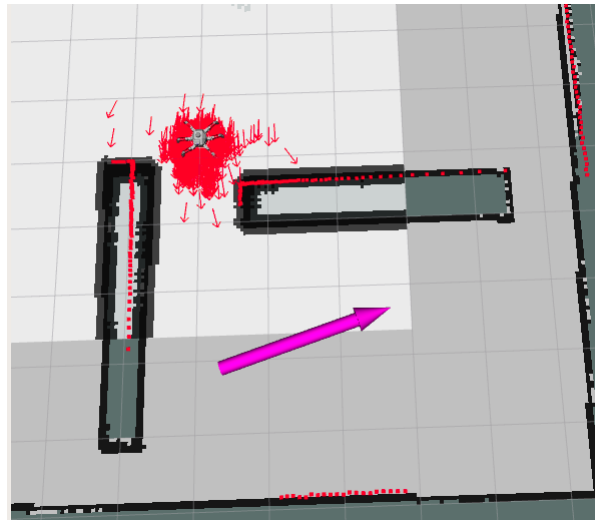


FIGURE 5.5 – Le test de navigation avec le robot ArIA Inspector.

Résultats de la navigation

Les résultats montrent la précision de la localisation et de la navigation. AMCL montre un déphasage de 0,2 mètre, dû à une variation de la vitesse du robot. Les résultats montrent une erreur maximale de 15 cm sur l'axe des abscisses à la fin de l'expérimentation, voir figure (5.6). Concernant l'axe des ordonnées présenté sur la figure (5.6), nous avons une erreur de 4cm entre la localisation AMCL et l'erreur de la vérité terrain.

Au sujet des résultats sur l'orientation Z, figure (5.6), on obtient une bonne correspondance entre la vérité terrain et l'AMCL au cours du processus, mais à la fin on obtient

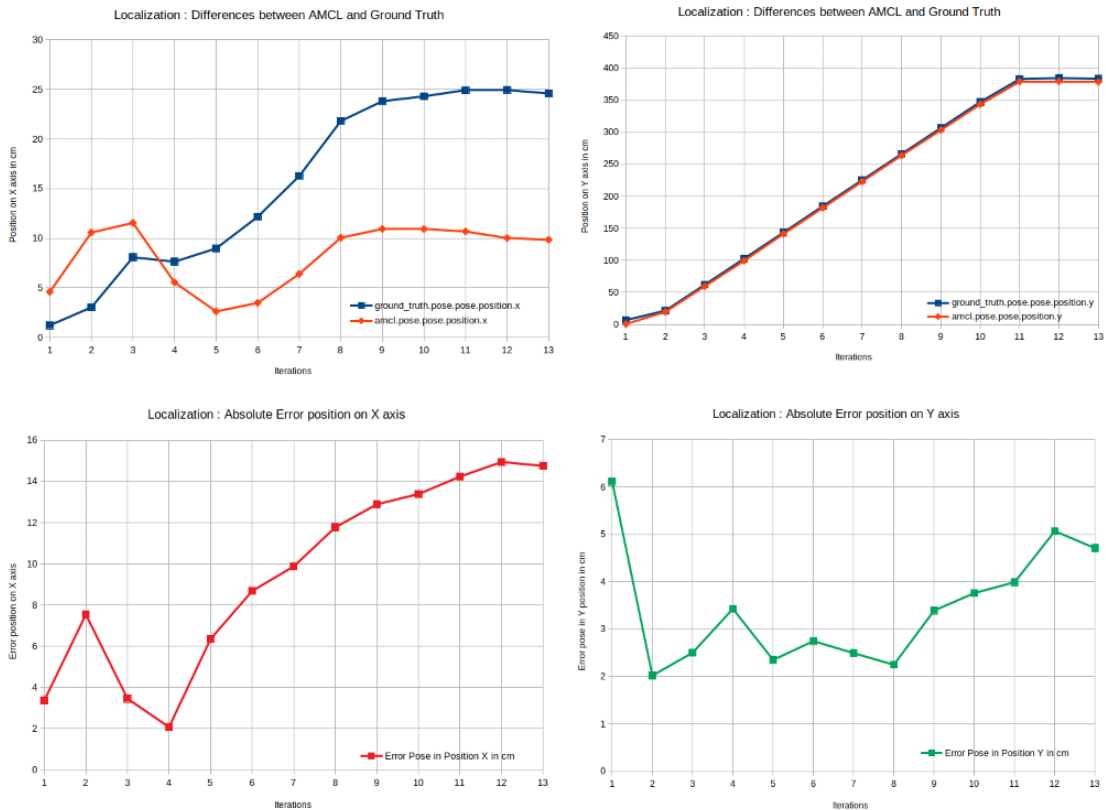


FIGURE 5.6 – De gauche à droite : 1) le tableau de comparaison entre AMCL et la vérité terrain du robot sur la position sur l’axe X, 2) sur l’axe Y, 3) L’erreur absolue de la position sur l’axe X, 4) sur l’axe Z .

une erreur cumulée due à l’action de rotation lorsque le robot atteint le point de navigation donnée, il s’agit d’une erreur cumulée sur l’axe d’orientation. Enfin, l’erreur due à la localisation sur un but simple est de 14 cm en abscisse (figure 5.6) et de 5 cm en ordonnée (figure 5.6). L’erreur est plus grande sur l’axe des x que l’axe y car l’oscillation de la démarche, la vitesse de démarche et la précision de l’algorithme peuvent influencer les résultats. Ces résultats montrent les bonnes performances du système de localisation sur un robot hexapode.

5.1.4 Expérimentations et résultats de l’asservissement visuel

Protocole d’expérimentation de l’asservissement visuel

Un robot statique (Turtlebot) aura un marqueur aruco attaché et sera la cible à atteindre. ArIA Inspector devra atteindre la cible et elle sera visible au début de la simula-

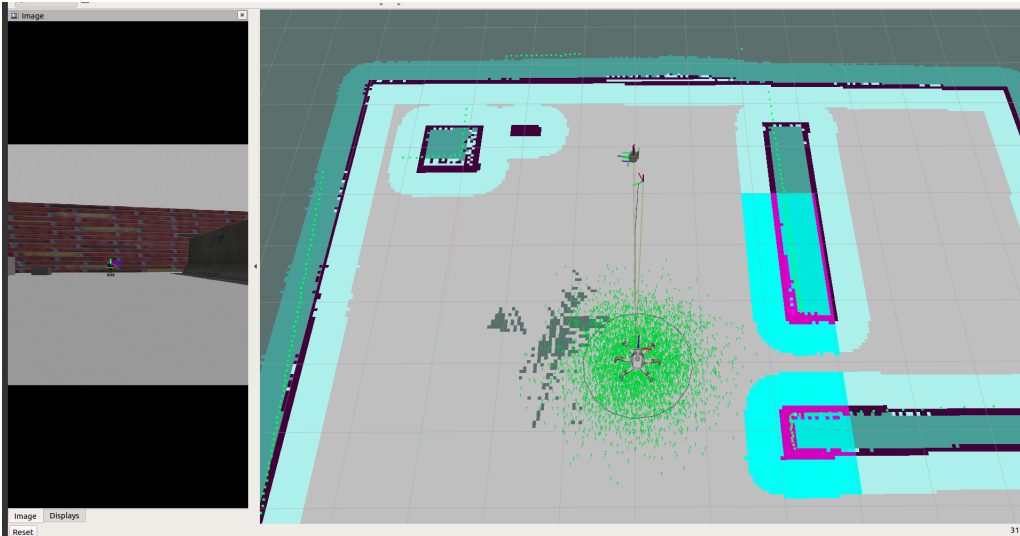


FIGURE 5.7 – L’asservissement visuel réalisé avec un QRcode.

tion. Nous exposons ici les méthodes d’asservissement par 10 itérations pour démontrer les performances de l’approche. Les résultats des essais d’asservissement visuel sont présentés et discutés.

Résultats de l’asservissement visuel

L’asservissement visuel est testé sur 10 essais. À droite de l’image se trouve une fenêtre Rviz avec ArIA au centre, le robot (Turtlebot) à droite et le repère présumé du marqueur aruco au-dessus d’Aria, notre robot hexapode. La convention veut que le plan image soit en x, y tandis que l’axe z représente la profondeur de champ. Cependant, dans l’univers simulé, le plan x, y correspond au sol et l’axe z mesure la hauteur. Dans la figure 5.7, la connexion en ligne jaune, représente la transformation homogène, qui relie le robot à sa cible : le marqueur aruco. Le traitement du repère aruco permet de publier la pose et l’orientation du marqueur. Un lien vers la vidéo est disponible : <https://youtu.be/a1iWFX1vXtM>. La vidéo décrit le procédé d’asservissement visuel en utilisant un marqueur aruco. On peut observer une certaine difficulté à détecter le marqueur lorsque ArIA est à 2 mètres de la cible. La détection du marqueur par la caméra simulé peut dépendre de plusieurs facteurs : les réglages de la caméra dans Gazebo, les vibrations de la démarche ou l’orientation de la caméra à cause du corps.

La problématique est de pouvoir maintenir le percept visuel sur un robot marcheur. La démarche d’un robot marcheur cause des variations dans la position du corps et donc,

des vibrations sur l'image. L'enjeu de cette problématique est de maintenir un contact permanent avec la cible pour réaliser l'asservissement visuel. Le robot doit être capable de générer des destinations successives à une certaine distance de sa cible. Lors de certaines trajectoires difficiles comme des rotations brusques, la cible peut être perdue. Dans ce cas, le robot doit se déplacer vers le dernier point de destination généré. Si, une fois arrivé à destination, ce dernier n'a toujours pas la cible en vue, alors un comportement d'exploration s'enclenche pour retrouver la cible. Pour le moment, ce comportement se limite à une loi de commande simple : un tour complet sur lui-même pour trouver la cible suivie d'une progression en marche vers celle-ci.

La problématique était de maintenir la détection du percept visuel sur un robot marcheur. Un essai unitaire a été réalisé pour vérifier les réglages et identifier les axes d'amélioration. Une série de 10 essais permet de mettre en lumière la répétabilité du procédé. Sur le premier essai, les résultats montrent la trajectoire globale du robot sur les axes x et y sur la figure (5.8), la trajectoire par rapport à la figure cible (5.9), et l'erreur relative sur l'axe des x et l'axe des y sont présentées sur la figure (5.10). Concernant l'asservissement visuel, le robot atteint la cible et s'arrête à 1,22 mètre de distance du marqueur aruco. L'erreur sur l'axe x correspond au centrage et l'axe y correspond à la distance entre le robot et la cible. Cette distance est expliquée aussi par les différents paramètres de navigation (l'empreinte de navigation du robot) pour éviter les collisions. Les résultats montrent que le processus peut être effectué sur une moyenne de 3 minutes.

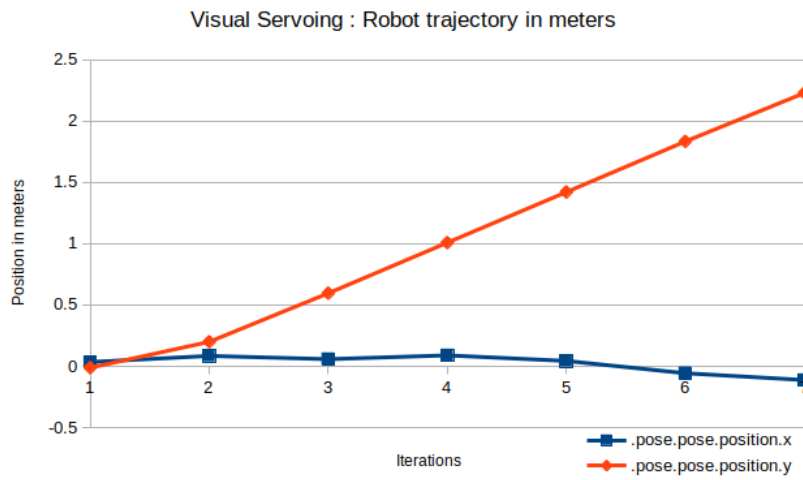


FIGURE 5.8 – Asservissement visuel : Trajectoire du robot sur les axes X et Y en mètres.

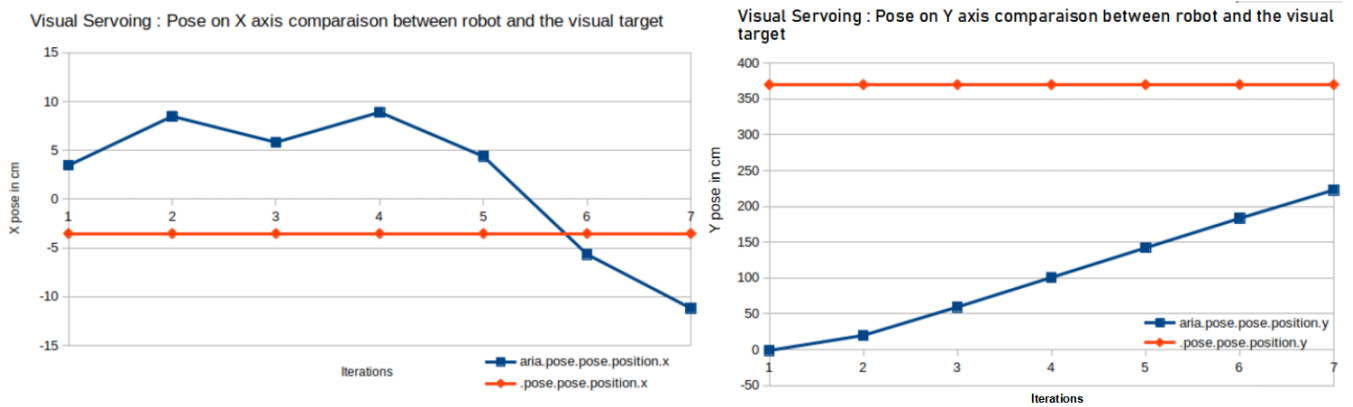


FIGURE 5.9 – Asservissement visuel : Trajectoire du robot par rapport à la cible sur l'axe X et Y en cm.

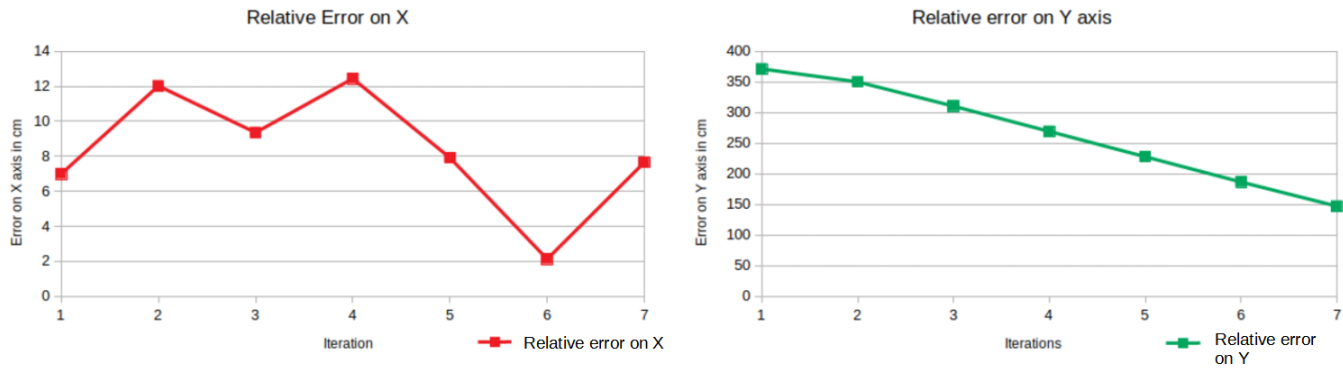


FIGURE 5.10 – Asservissement visuel : Erreur relative du robot sur les axes X et Y en cm.

Essai sur 10 itérations.

Les résultats sur de la durée d'exécution (5.11) montrent une performance maximale à 51 secondes pour atteindre la cible (Essai 6) avec une erreur absolue (figure (5.12)) de -0,054 mètre sur l'axe x et -1,19 mètre sur l'axe y (Essai 6).

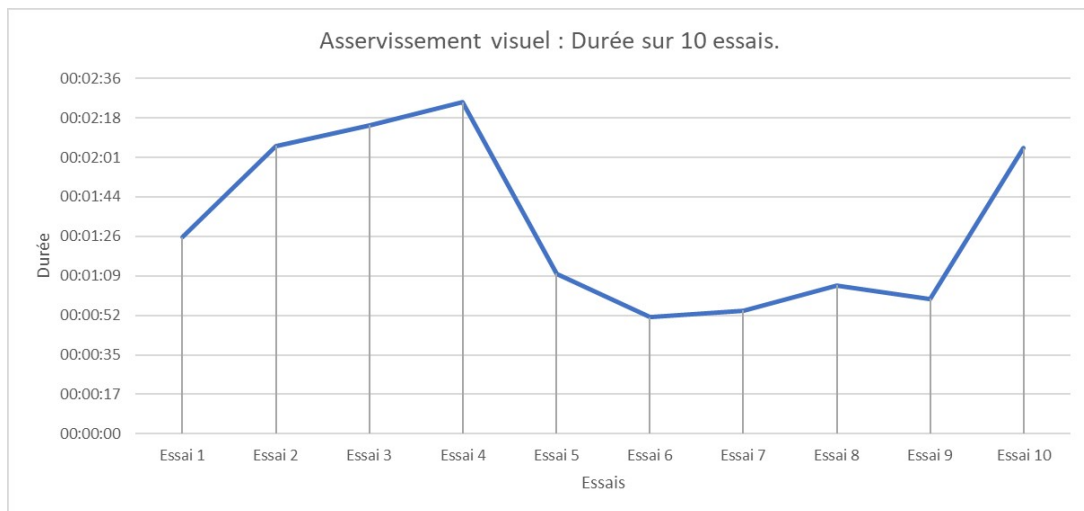


FIGURE 5.11 – Asservissement visuel : Répartition de la durée d'asservissement.

L'erreur absolue moyenne est de -0,0018 m sur l'axe x et -1.34 m sur l'axe y. Les paramètres de navigation influent aussi l'asservissement visuel, laissant une distance de sécurité entre le robot et un obstacle.

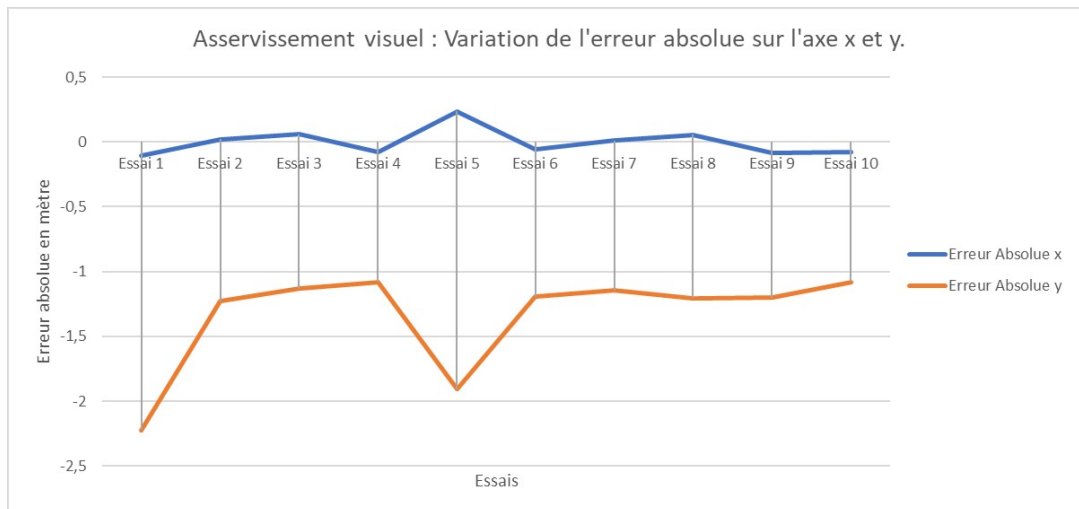


FIGURE 5.12 – Asservissement visuel : Répartition de l'erreur absolue.

Même si le procédé s'est bien déroulé sur les 10 essais. On peut noter une précision plus faible que les autres sur l'essai 1 : une erreur absolue de -2,22 mètres sur l'axe y. Ce phénomène est observé lorsque le percept visuel, le marqueur aruco (QRcode), est détecté par intermittence, entraînant une trajectoire saccadée. Les paramètres visuels jouent un rôle important et dépendent des réglages, la calibration et paramètres intrinsèques de la caméra (la distance focale, les facteurs d'agrandissement de l'image, les coordonnées de la projection du centre optique de la caméra sur le plan image). Elle peut aussi être améliorée d'un point de vue algorithmique en conditionnant un comportement basé sur l'erreur absolue (la distance entre le point de référence du robot asservi et le point de référence du QRcode).

5.1.5 Entraînement et résultats de YOLO V5

Base de données et entraînement de YOLO v5

La base de données d'entraînement est présentée sur la Figure (5.1) et comporte des défauts industriels dans le milieu métallurgique (de la tache de graisse sur un tuyau jusqu'aux défauts de soudure). La répartition de la base de données est inégale sur certaines catégories (Crease, Rolled pit), malgré un processus d'augmentation d'image (variation du bruit, contraste et luminosité).

La base de donnée de soudure est séparé en 3 familles de défauts : les défauts du bâtiment, les défauts de travail sur bobine de métal et les défauts liés à la soudure. Nous avons répertorié un défaut dans cette catégorie :

- Crack : fissure encastrée le mur.

Les défauts de travail sur la bobine sont les suivants :

- Crease : Pli sur le métal,
- Crescent gap : défaut en demi-cercle lié à la découpe,
- Inclusion : défauts de surface de différentes formes (p. ex., forme d'écaille de poisson) qui peuvent être lâches et faciles à tomber ou à presser dans le métal,
- Oil spot : contamination causée par un lubrifiant mécanique qui affecte l'apparence du produit,
- Punching hole : bandes d'acier avec des trous de perforation supplémentaires indésirables causés par une défaillance mécanique,
- Rolled pit : des renflements ou des fosses périodiques à la surface du métal, souvent causés par des dommages ou à la tension du rouleau de métal,
- Silk spot : plaque ondulatoire à la surface souvent causée par la pression ou la température inégale d'un rouleau,
- Waist folding : plis ressemblant à des rides causés par un faible taux de carbone,
- Water spot : les tâches qui se produisent lors du séchage du métal pendant la production,
- Welding line : une ligne de soudure qui se produit lorsqu'une bande d'acier est changée,
- Crazing : produit un réseau de fines fissures à la surface d'un matériau,
- Inclusion : défauts de surface de différentes formes (p. ex., forme d'écaille de poisson) qui peuvent être lâches et faciles à tomber ou à presser dans le métal,

- Patches : métal meuble qui a été roulé dans la surface de l’acier,
- Pitted surface : type de corrosion qui se produit dans les matériaux qui ont des films protecteurs,
- Scratches : Rayure sur le métal.

Les défauts de soudure sont les suivants :

- Burn through : température trop haute / métal brûlé,
- Contamination : corrosion galvanique entre des particules de fer (poussière de fer) déposées sur la surface,
- Good weld : soudure correctement réalisée,
- Lack of fusion : Lorsque le matériau de soudure ne fusionne pas correctement, ce qui entraîne un joint faible,
- Lack of penetration : Une mauvaise pénétration de la soudure se produit quand le métal soudé ne pénètre pas complètement dans le joint et entraîne un joint faible,
- Misalignment : Relève des imperfections géométriques obtenues dans le processus de soudage.

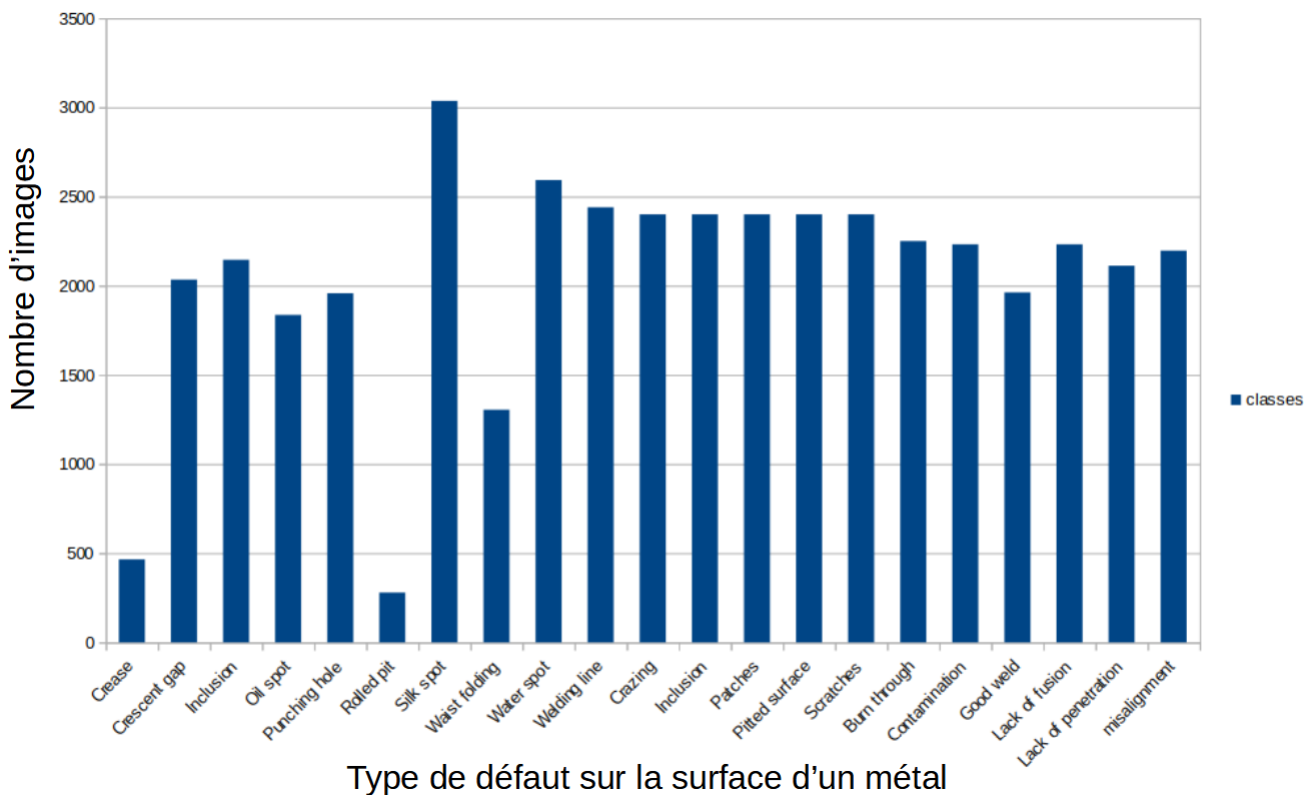


TABLE 5.1 – Les différents défauts à détecter.

La base de données a été augmentée par un filtre de bruit ($\sigma = 0.255$), un filtre de flou, un filtre sépia, un filtre vert, bleu et rouge, un filtre sombre ('Mexican hat' composé d'un filtre Gaussien et Laplacien) et enfin un filtre de netteté sur l'ensemble de la base de données avant le procédé d'entraînement. Après plusieurs tests, l'architecture de YOLO 5(S) a été initialisée avec 22 classes, 8 couches de convolution et 4 de concaténation. Les fenêtres ("anchors") sont au nombre de 3 et déterminées par une procédure d'algorithme de regroupement, par "cluster" ("k-means"). L'architecture de YOLO 5(S) a été initialisée avec les paramètres figurant dans l'annexe F.

Résultats de l'entraînement de YOLO V5

Dans cette partie, nous présentons les résultats de l'entraînement de YOLO v5. Pour mesurer les résultats d'entraînement, nous avons appliqué plusieurs métriques : la matrice de confusion, le score F1, le corrélogramme, la précision, le rappel ('recall'), le diagramme de précision et l'erreur sur la boîte englobante. Nous présentons sur la figure 5.13, en A) la base de données utilisée pour l'entraînement de l'algorithme (valeur 0 pour fissure), en B) l'empreinte et la récurrence des défauts détectés, en C) la récurrence de la position de la fenêtre de détection et en D) la taille des défauts.

La matrice de confusion est présentée sur le graphique 5.14. La matrice de confusion est utilisée ici pour définir la performance de l'algorithme par une comparaison des données réelles pour une variable cible et celle prédite par le modèle. On peut voir que les résultats pour les fissures (label 0) obtiennent un score de précision de 83% et de 90% pour le défaut d'inclusion. En revanche, la catégorie scratches n'a pas été testée durant l'entraînement.

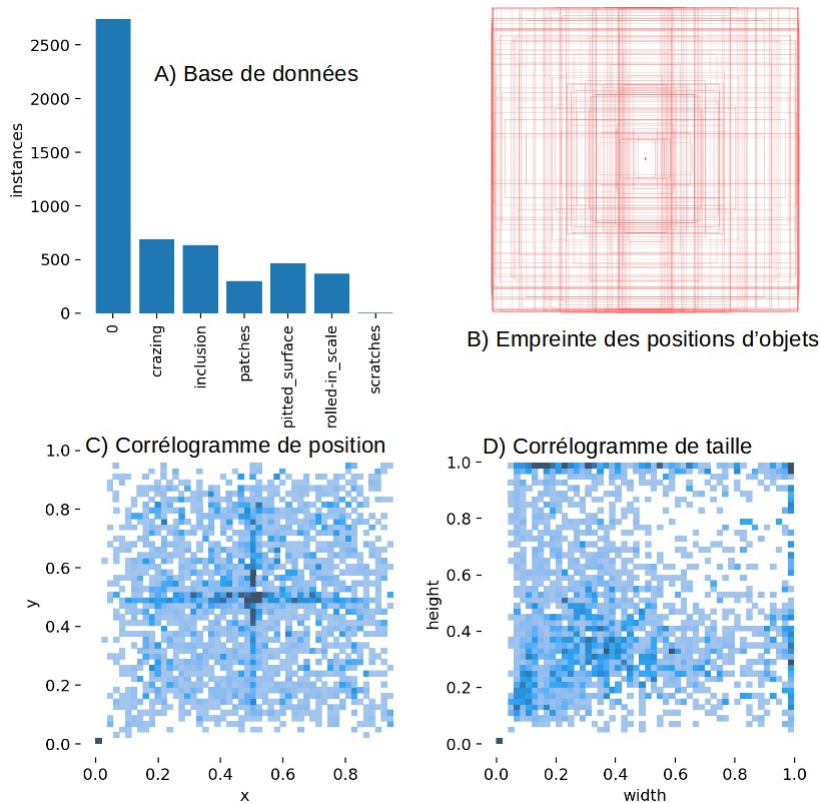


FIGURE 5.13 – La matrice de corrélation suite à l'entraînement. A) La base de données utilisées, B) Empreinte des positions de l'objet, C) Corrélogramme de position et D) Corrélogramme de taille.

Le score F1 (ou F-measure en anglais) est une moyenne de la précision et du rappel. Le graphique 5.15 indique les scores F1 des différentes classes de la base de données. Les résultats sont plutôt moyens pour l'ensemble des classes ($F1 = 0.68$). Les résultats des autres classes avoisinent 80%.

Le corrélogramme permet de mettre en évidence une ou plusieurs corrélations entre les séries de données. Il permet, en d'autres termes, d'analyser une relation entre chaque paire de variables numériques d'une matrice (ici la hauteur et la largeur d'une image et l'occurrence des points d'intérêts sur l'image). Sur le graphique 5.16 (B) et (C), nous pouvons voir qu'il y a une forte concentration des points d'intérêts vers le milieu de l'image, ce qui veut dire que la majorité des défauts de la base sont centrés dans l'image. Sur le graphique (D), on distingue une majorité de mesure sur le côté gauche formant un carré de 0.4 width sur 0.4 height. On peut en déduire que les défauts de la base font 0.4×0.4 en taille.

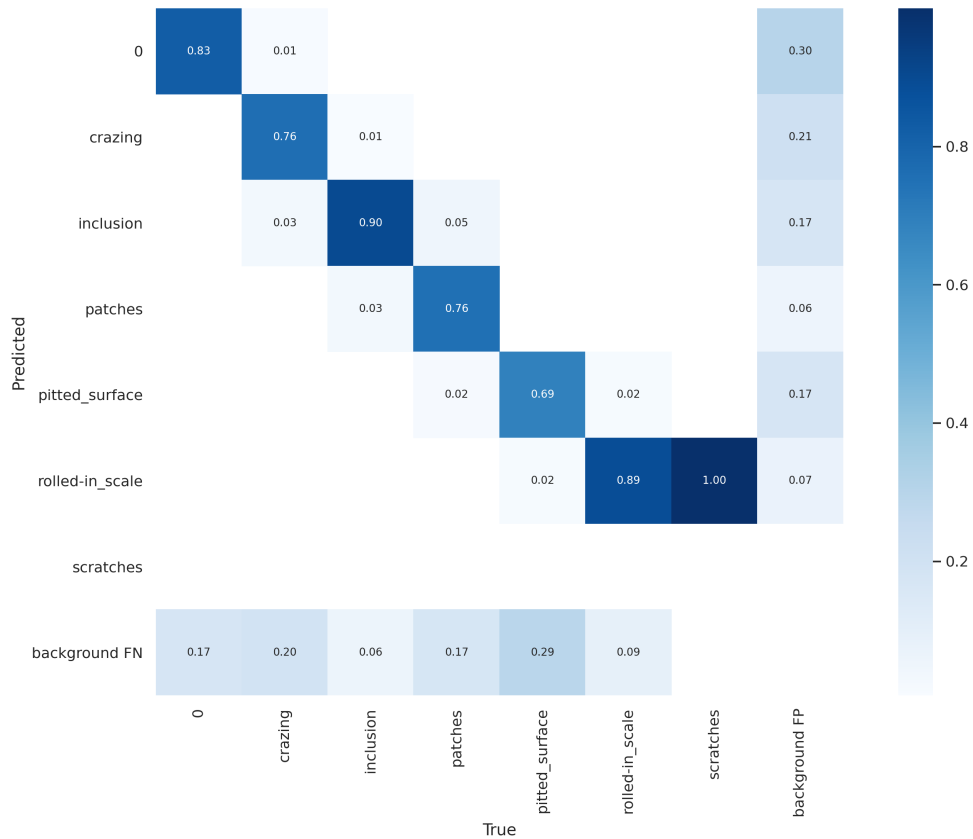


FIGURE 5.14 – Matrice de confusion de la base de données de défauts industriels.

La précision indique le rapport entre les prévisions positives correctes et le nombre total de prévisions positives. Ce paramètre répond donc à la question suivante : sur tous les enregistrements positifs prédits, combien sont réellement positifs ? Les résultats de la précision sont présentés figure 5.17. Les différentes classes ont un bon score de précision qui avoisine les 90% pour la majorité des classes.

Le rappel (ou recall en anglais) est un paramètre qui permet de mesurer le nombre de prévisions positives correctes sur le nombre total de données positives. Il permet de répondre à la question suivante : sur tous les enregistrements positifs, combien ont été correctement prédits ? Les résultats du rappel sont présentés figure 5.18. Les différentes classes des scores avoisinent globalement les 80 % de défauts positifs prédits avec exactitude.

Le diagramme Précision Rappel (ou precision recall en anglais) conjugue les deux indicateurs précision (prédictions correctement réalisées) et rappel, afin d'obtenir un indicateur de performance concret pour le modèle par leur complémentarité. Le graphique des

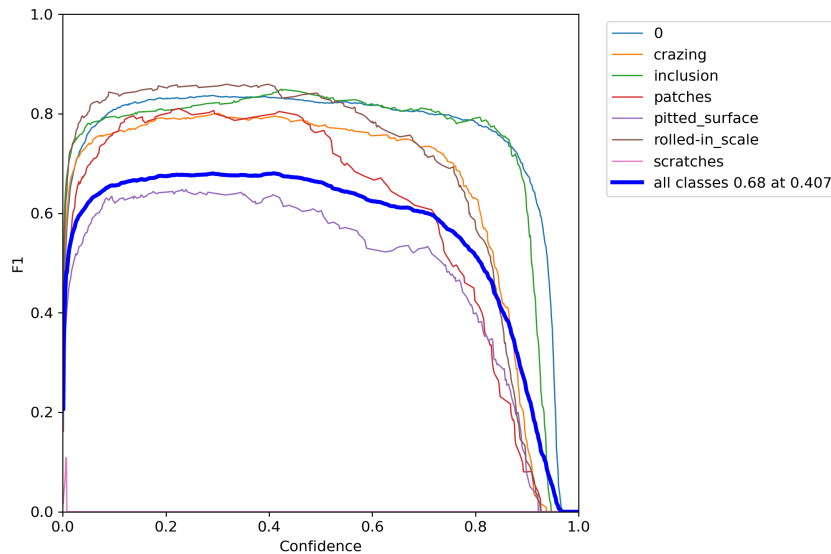


FIGURE 5.15 – Résultats de la courbe F1 suite à l'entraînement.

résultats Précision-Rappel 5.19 montre que les résultats des différentes classes avoisinent les 80% pour les fissures (objet 0), "crazing", l'inclusion, les patches et les "rolled in scale". En revanche, les "pitted-surfaces" ont le score le plus faible : 59 %.

Les performances peuvent être évaluées sur les fractionnements de données d'entraînement, de validation ou de test. L'évaluation permet, dans notre cas, de comparer les résultats d'entraînement et la validation, vérifiant l'algorithme sur une sélection d'images non diffusées pendant l'entraînement. Les résultats sont montrés ci-dessous sur le graphique 5.20. Nous regroupons ici différentes métriques utilisées afin d'avoir une vue d'ensemble des performances de l'entraînement. Les métriques employées ici sont la perte (ou erreur) de la boîte englobante ("box loss"), l'erreur de l'objet ("obj loss"), l'erreur de classification ("cls loss"), la précision ("precision"), le rappel ("recall") et la précision moyenne ("mAP 0.5") avec union des intersections (IoU).

L'erreur sur la boîte englobante ("box loss") correspond à la perte de régression du cadre de sélection. Elle est calculée par l'erreur quadratique moyenne. L'erreur d'objet ("obj loss") est la confiance de la présence d'objet est la perte d'objectivité. Elle est calculée par l'entropie croisée binaire. L'erreur de classification ("cls loss") est calculé par l'entropie croisée. La précision moyenne ("mAP_{0.5}") est la précision moyenne (mAP) au seuil IoU (Intersection over Union) de 0,5. mAP_{0,5} : 0,95 est le mAP moyen sur différents seuils IoU, allant de 0,5 à 0,95.

Globalement, les résultats de l'entraînement correspondent avec les résultats de la

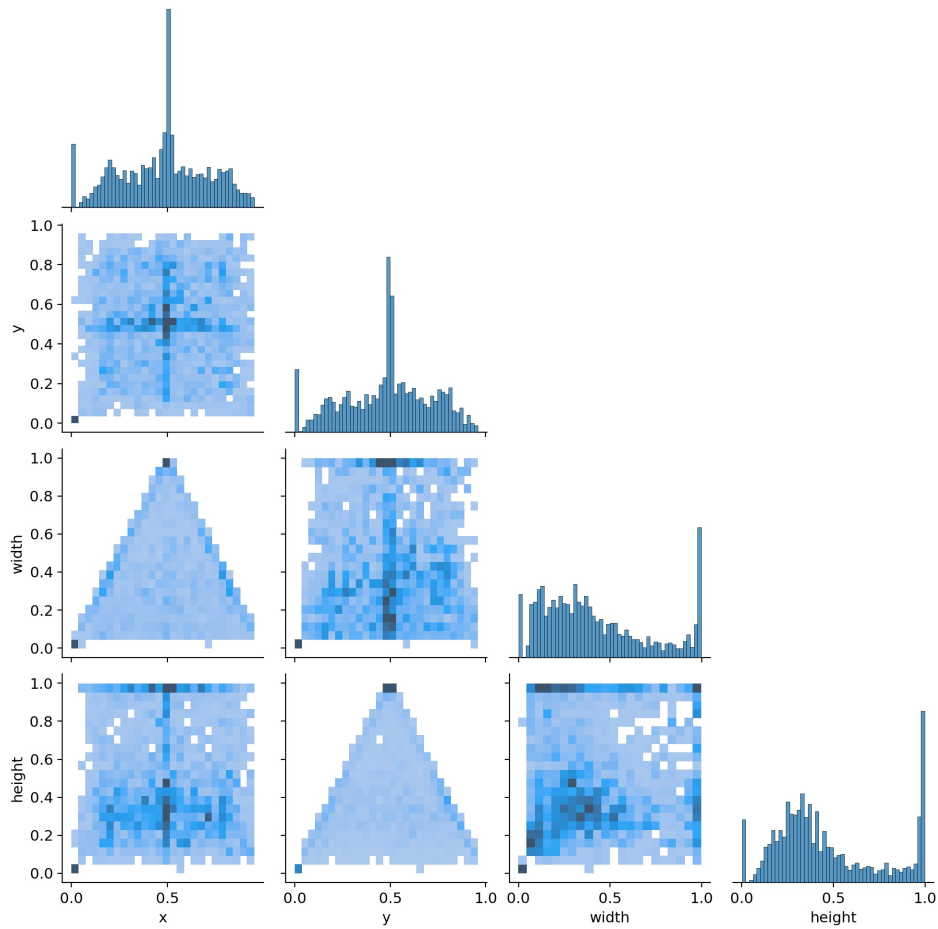


FIGURE 5.16 – Résultat du corrélogramme des différentes catégories de l’entraînement avec l’occurrence des classes en fonction de la largeur et la hauteur.

validation. Au vu des résultats, il ne semble pas avoir de problème de sur-apprentissage. À la suite de l’entraînement, une inférence est générée au format PyTorch (.pt).

Les résultats de la détection visuelle lors de l’étape de validation sont présentés figure 5.21 et 5.22. Sur la figure 5.21 et 5.22, on peut voir que les différents défauts sont correctement identifiés et conforme aux attentes des procédés.

De manière générale, la précision dépasse les 80% et nous avons une corrélation correcte entre les données d’entraînement et de validation. Au regard des résultats, l’inférence sera implémentée dans le GPU embarqué sous un format ROS pour permettre la communication des données à l’ensemble des services ROS.

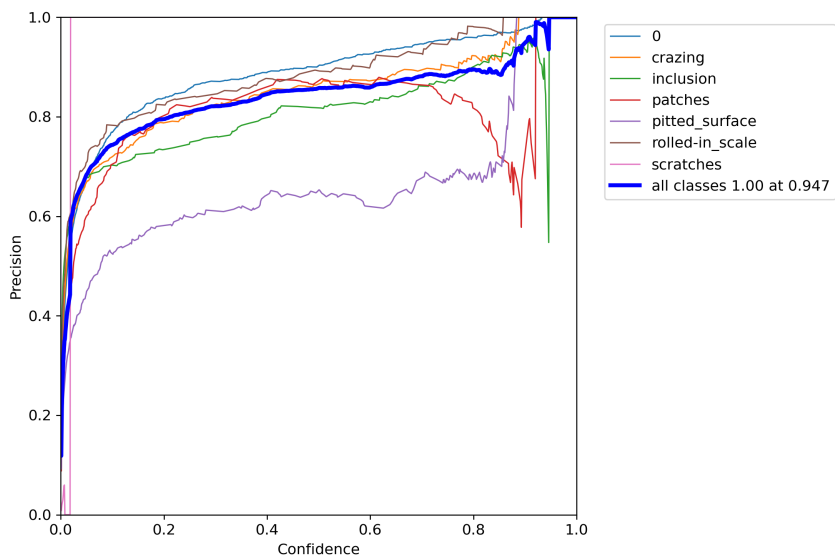


FIGURE 5.17 – Résultat de la précision des différentes catégories de l’entraînement correctement prédites.

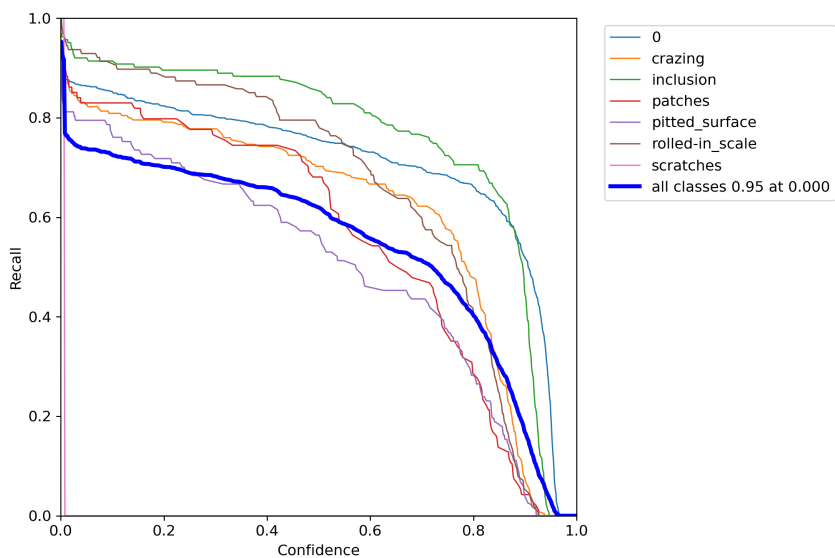


FIGURE 5.18 – Courbe de rappel ("recall") des différentes classes de l’entraînement en fonction de l’indice de confiance.

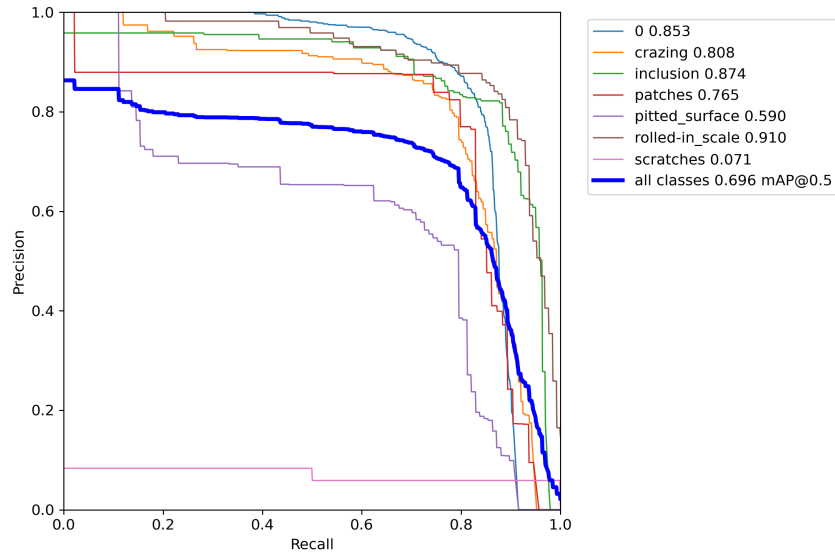


FIGURE 5.19 – Résultats de la courbe Précision Rappel sur la base de données de défauts industriels.

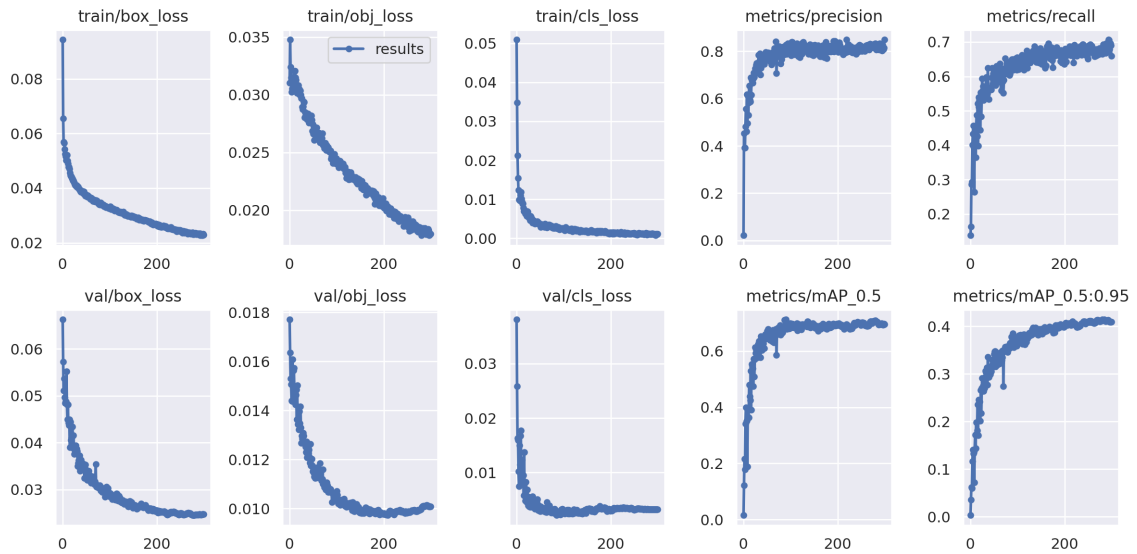


FIGURE 5.20 – Résultats de la métrique sur l'entraînement de YOLO v5 en considérant toutes les classes.

5.1.6 Expérimentation et résultat de l'asservissement chimique

Expérimentation de l'asservissement chimique

Nous allons tester l'intégration des capteurs de gaz dans ROS ainsi que la possibilité de créer un environnement et d'y charger des données de dynamique des fluides et simuler

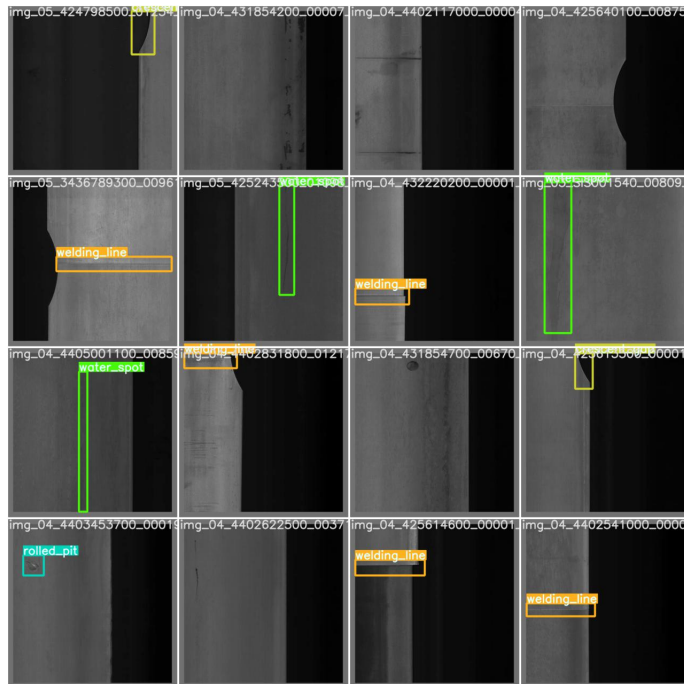


FIGURE 5.21 – Résultats visuels de l'étape de validation concernant l'entraînement de YOLO v5 sur la base de données de défauts industriels, ici les défauts sont variés et cible le métal.

un écoulement de gaz. Ce paquet est basé sur l'utilisation de Stage plutôt que de Gazebo pour la simulation environnementale, voir figure 5.23. Pour la première expérience, un environnement de 10×6 mètres avec un seul obstacle (Expérimentation A), voir figure 5.23. L'intérêt de cette expérimentation est de démontrer la faisabilité de 1) la modélisation d'un flux gazeux et 2) l'asservissement chimique d'un robot hexapode dans un milieu contraint d'un obstacle.

La deuxième expérimentation se déroule dans un labyrinthe (Expérimentation B). Notre objectif est de démontrer la capacité de traverser un labyrinthe et d'atteindre la source même si le cycle de simulation atteint la fin de la boucle de données de simulation de particules. Pendant cette période critique, la simulation ne diffuse plus le panache chimique, l'objectif est de valider le système de récupération qui vise à retrouver la trace du percept chimique (le panache) et de reprendre sa recherche, voir figure (5.24). Cet environnement est un cas d'environnement contraint par la présence d'un couloir de 1 mètre de largeur.

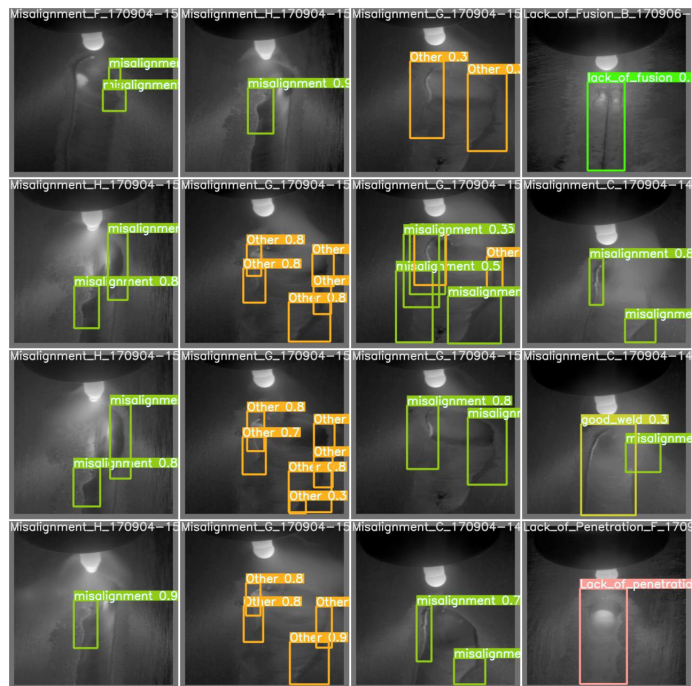


FIGURE 5.22 – Résultats visuels de l'étape de validation concernant l'entraînement de YOLO v5 sur la base de données de défauts industriels, ici les défauts de soudure.

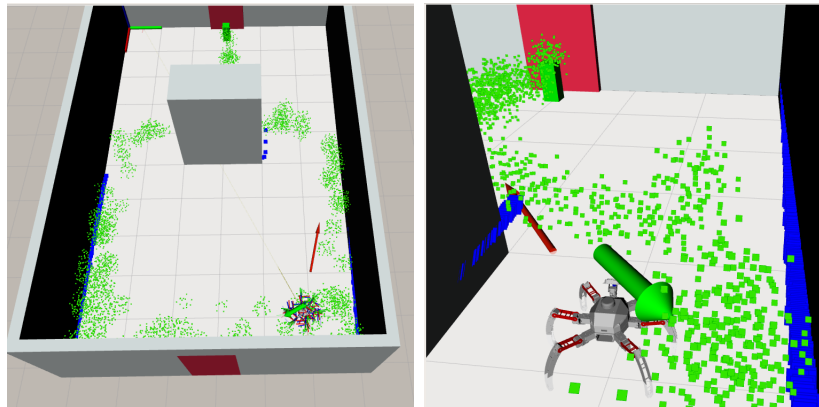


FIGURE 5.23 – Visualisation RViz de l'environnement "obstacle central" avec la simulation de propagation du gaz.

Résultat des algorithmes de l'asservissement chimique

L'asservissement chimique est testé en utilisant plusieurs méthodes de recherche de source de gaz ("Grid"), "Surge Cast", "Spiral", "Surge Spiral", "Particule filter". Le test est réalisé en utilisant ROS et RViz, figure 5.25. Sur la Figure 5.25, les particules vertes représentent la dispersion du gaz provenant du cube vert en haut de la pièce. Le rectangle

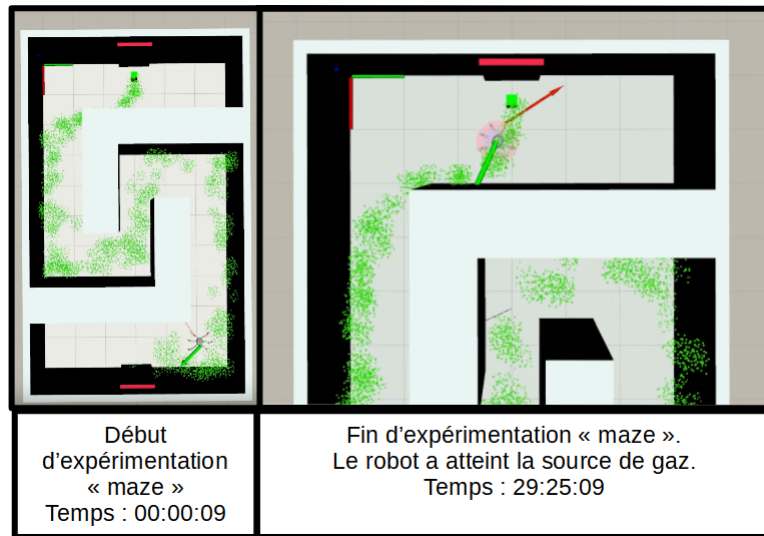


FIGURE 5.24 – Visualisation RViz de l'environnement "maze" avec la simulation de la propagation du gaz. Sur l'image de gauche, le début de l'expérimentation avec le robot au point de départ et la source de gaz matérialisée par une boîte verte. Sur l'image de droite, le robot a atteint la source de propagation du gaz (boîte verte).

rouge représente une ouverture vers un flux d'air. Les carrés bleus illustrent la captation laser, permettant au robot de se localiser par rapport à celui-ci. La carte est préalablement enregistrée. L'asservissement chimique est testé sur 10 essais dans un environnement comportant un obstacle central. La problématique est que l'obstacle provoque un phénomène de siphon sur la trajectoire du flux gazeux. On suppose que cette difficulté peut entraîner des erreurs et des singularités dans la trajectoire et ainsi provoquer une perte de suivi du flux gazeux.

L'asservissement chimique est réalisé en visualisation (RViz) et les essais sont séparés en 3 parties. Le premier essai est un test de durée d'asservissement chimique sur les algorithmes de recherches de source de gaz ("Grid", "Surge-Cast", "Spiral", "Surge-Spiral", "Particule filter"). Un essai unitaire est réalisé pour vérifier les réglages. Une série de 10 essais permet quantifier la répétabilité du procédé. Le but est de déterminer l'algorithme qui converge le plus rapidement.

Résultat des algorithmes chimiques

Dans l'environnement avec un obstacle, nous testons différents algorithmes afin de déterminer le plus adapté pour un robot marcheur. Sur la figure (5.26), nous pouvons voir que la méthode qui convient le mieux est l'algorithme "Grid" avec un asservissement

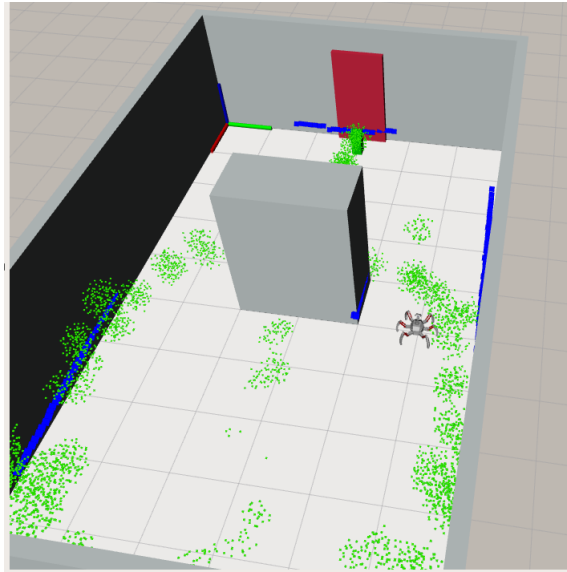


FIGURE 5.25 – Le test d’asservissement chimique sous RViz en utilisant l’environnement Gaden [176].

à 3 minutes et 37 secondes.

- Lien vidéo de la méthode "Grid" : https://youtu.be/BIIdVt_h1nuM.
- Lien vidéo de la méthode "Surge-Cast" : <https://youtu.be/e9pKm3Gx-8Y>.
- Lien vidéo de la méthode "Particle-Filter" : <https://youtu.be/bGU2j8QVu2s>.
- Lien vidéo de la méthode "Spiral" : <https://youtu.be/xsEtnw50pwo>.

La méthode "Grid" démarque des autres sur le fonctionnement de sa loi de commande. Cet algorithme découpe la carte en grille (d’où son nom) et utilise la méthode des proches voisins pour attribuer un poids, une probabilité de présence d’un panache gazeux, initialement considéré comme vide. La méthode prend en compte la direction du flux d’air et la concentration ppm des capteurs de gaz. Durant l’essai, il y a eu peu de points d’arrêt en dehors du panache. La trajectoire reste stable même si une perte de percept est déclenché par la fin de la boucle de propagation du gaz (en 3min28). Les résultats sur le temps figure 5.27 montrent une durée maximale de 5 minutes et 51 secondes pour atteindre la cible (Essai 6) avec une erreur absolue de -0,09 mètre sur l’axe x et 0,61 mètre sur l’axe y (Essai 6). En revanche, les erreurs absolues sur les essais 7 et 9 sont anormalement élevés :

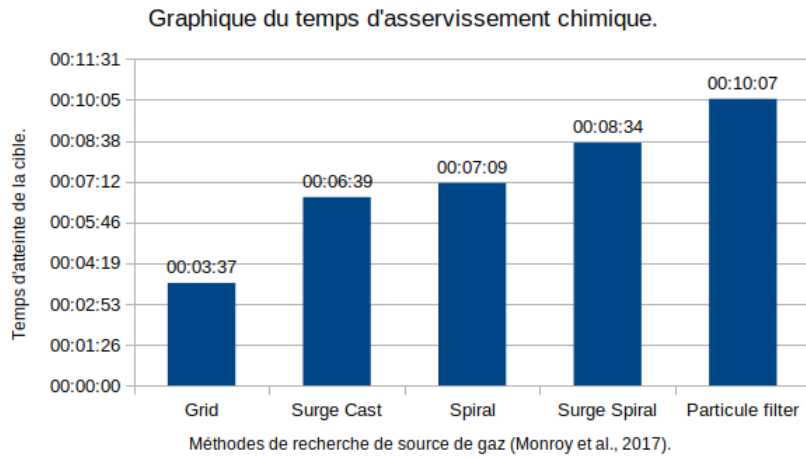


FIGURE 5.26 – Essai des différentes méthodes de recherche de source de gaz avec des capteurs de type métal oxydes simulés.

avec 2,81 mètres (essai 7) et 1,62 mètre (essai 9) pour l'axe des x et 0,7 mètre (essai 7) et 0,16 mètre (essai 9).

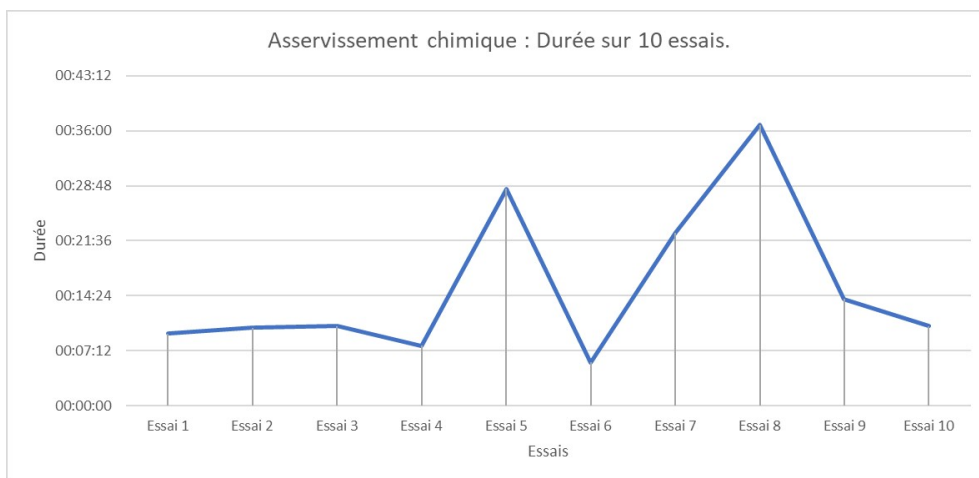


FIGURE 5.27 – Asservissement chimique : Répartition de la durée d'asservissement.

Globalement, l'asservissement chimique a pu être opéré de manière stable. L'essai 7 et l'essai 9, ont mis en évidence deux phénomènes à corriger. Durant l'essai 9, lorsque le robot arrive au niveau de l'obstacle, la patte avant gauche a tendance à se coincer au niveau de l'obstacle. Cela est dû à un réglage de l'empreinte du robot sur ROS. Une solution serait d'utiliser l'espace de démarche (généré par le contrôleur) ou autrement dit l'espace de travail des positions des fins d'effecteurs comme coordonnées d'empreintes du

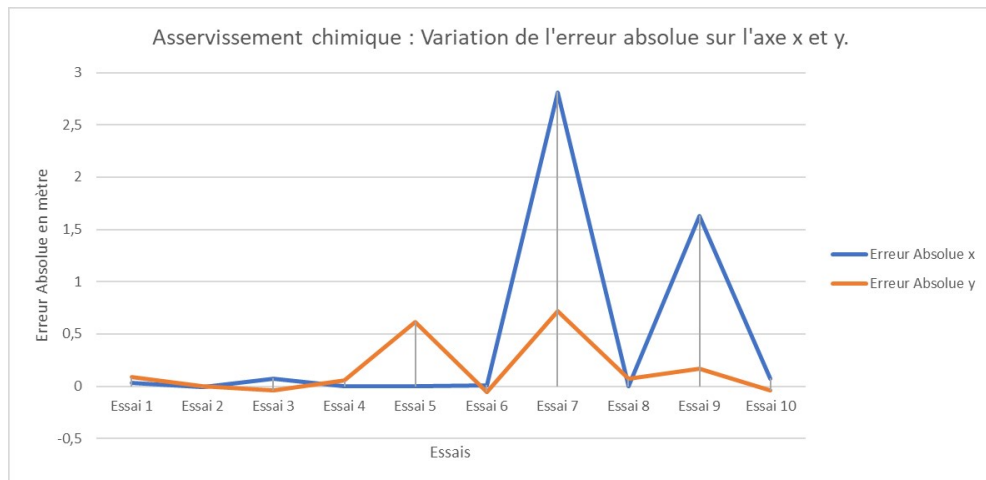


FIGURE 5.28 – Asservissement chimique : Répartition de l'erreur absolue.

robot afin d'appréhender ce problème. Lors de l'essai 7, dans le même cas de figure que l'essai 9, le robot part dans une direction opposé à la direction du flux d'air et à l'opposé de l'obstacle. Sur le Test 7 et 9, le robot a débuté l'essai 2 mètres à gauche du départ initiale. Le flux était alors en dans le sens inverse et dirige donc le robot dans la direction opposé à la source. Il en résulte que le robot éprouve des difficultés à rejoindre le panache de gaz, dû à une distance l'éloignant du panache trop important.

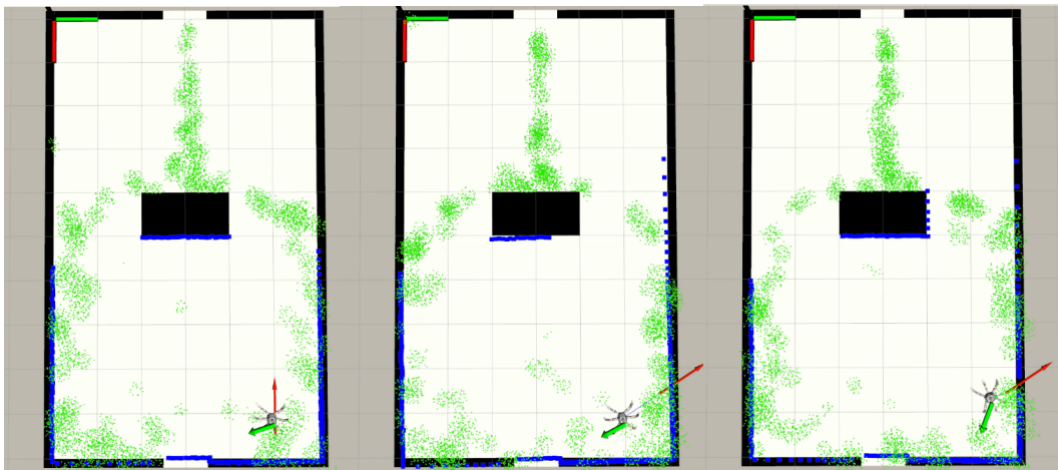


FIGURE 5.29 – Asservissement chimique : obstacle central.

Durant le procédé d'asservissement chimique (Figure 5.29), la flèche verte représente le vecteur de direction du flux d'air et la flèche rouge est le vecteur de commande, générant les points de navigation à partir du système de l'algorithme d'asservissement.

Conclusion

Les essais en simulation ont démontré qu'un asservissement chimique avec une navigation autonome pour un robot hexapode est réalisable en simulation. Les erreurs de trajectoire de la part du robot sont principalement dues à un point de navigation excentré de la trajectoire du panache. Ce phénomène peut entraîner des trajectoires aller-retours en dehors du flux gazeux. Ce phénomène se produit principalement lorsque le robot réalise une trajectoire proche d'un obstacle. Cette problématique peut apparaître comme un conflit entre les paramètres de coût pour la planification de trajectoire et l'asservissement chimique. La solution la plus simple est d'affiner le réglage des paramètres de coût. Une solution serait de trouver la trajectoire idéale réalisant le compromis entre évitement d'obstacle et suivi du percept chimique, en générant des primitives de mouvement sûr dans l'environnement. Et ainsi, générer toutes les trajectoires possibles. Ou bien, renforcer un comportement de recherche de gaz basé sur un algorithme apprentissage par renforcement.

5.2 Expérimentations sur le robot réel

Dans cette partie, nous allons étudier le comportement mécanique et la capacité de franchissement de ArIA Inspector en l'éprouvant sur des tests statiques, dynamique et des tests de franchissement d'obstacle en espace confiné confronté à un autre robot d'une architecture à chenille. Les hexapodes, et comme la plupart des robots marchants, ont des contraintes mécaniques plus fortes que pour des architectures de robot terrestres (roulant, chenille, etc.). Les 18 servomoteurs d'ArIA Inspector sont amenés à être fortement sollicités, que ce soit lors d'essais statiques, un maintien simple du robot, ou lors d'essais dynamiques, en déplacement holonome par exemple. Un effort trop important sur les servomoteurs entraîne leurs arrêts.

Les expérimentations réelles ont pour but de qualifier la robustesse de l'architecture des servomoteurs en termes de couple. L'enjeu de cette problématique est le maintien mécanique de l'hexapode pour des inspections avec de fortes sollicitations (durée, obstacle, etc.) évoqué lors de l'analyse des cas Chapitre 1 - Étude des éléments contextuels.

5.2.1 Expérimentation et résultats des essais statiques

Expérimentation des essais statiques

Notre hypothèse de travail est que la disposition du corps, la hauteur en particulier, peut davantage solliciter ou alléger les efforts mécaniques sur les servomoteurs. L'essai statique est réalisé avec 3 modes de gardes au sol. Le temps d'acquisition est de 20 secondes par essai. L'objectif est d'identifier une posture pour la marche sur terrain plat et une posture pour le franchissement d'obstacle. Nous allons évoquer et discuter les résultats des essais statiques, dynamiques et en situation réelle pour déterminer 1) si les efforts sont correctement répartis, 2) si les servomoteurs ne sont pas en surcharge et 3) si la conception du robot permet d'explorer un cas d'utilisation en milieu naval (n°1). Dans cette dernière hypothèse de travail, une attention particulière sera portée au franchissement d'obstacles. Dans ces essais, nous comparons la performance des MX28T avec d'autres modèles de servomoteurs, afin d'identifier, en plus des sollicitations d'efforts, des servomoteurs permettant de compenser les pics de couple. Les différentes positions sont montrées Figure 5.30. La position A est une garde au sol de 20 cm, la position B est une garde au sol de 15 cm et la position C, une garde au sol de 25 cm.

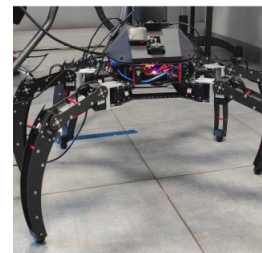
3 Configurations



Expérience 1
Position A
Garde au sol
20 cm



Expérience 2
Position B
Garde au sol
15 cm



Expérience 3
Position C
Garde au sol
25 cm

FIGURE 5.30 – Les 3 configurations différentes de garde au sol : Position A (20 cm), Position B (15 cm), Position C (25 cm).

Les objectifs sont d'évaluer l'impact de la garde au sol sur les efforts des servomoteurs et donc, repérer les articulations davantage sollicitées. L'idée est aussi de mettre en évidence les défauts de conception, d'équilibrage et pouvoir ressortir des informations pré-

cieuses pour améliorer l'architecture des pattes en effort mécanique et vitesse. De plus, les différentes positions représentent différents cas, la position C, par exemple, est utilisée pour franchir des obstacles, quant à la position A est la position nominale de marche.

Résultats en état statique

Le premier essai en position A (Figure 5.31) est réalisé avec un robot, une garde au sol de 20 cm. C'est la position standard déployée lors des différents essais. Les résultats montrent une sollicitation des servomoteurs d'environ 55% de leurs capacités. La moitié des servomoteurs sont concernés avec une dominance sur le côté droit.

Le deuxième essai en position B (Figure 5.32) met le robot en position B est une garde au sol de 15 cm. Cette posture permet théoriquement de réduire les efforts en abaissant le centre de gravité. En revanche, cette posture limite le franchissement d'obstacle. Dans cet essai, nous observons que seulement 4 servomoteurs sur 18 sont au-dessus des 50% de sollicitation et un seul dépasse le seuil maximum d'effort admissible.

Le troisième essai en position C (Figure 5.33) met le robot position C est une garde au sol de 25 cm. Cette posture permet théoriquement de franchir plus facilement les obstacles. En revanche, cette posture augmente les efforts sur les joints en augmentant la hauteur du centre de gravité. Les résultats montrent un effort plus important sur les servomoteurs des coxas.

Afin de faciliter la lisibilité, une liste de correspondance est illustrée entre le numéro du servomoteur et le nom de l'articulation. A, B et C correspondent respectivement aux la partie avant, milieu et arrière. L et R correspondent à la partie gauche ("Left") et droite ("Right") :

- AL COXA : 1
- AL FEMUR : 3
- AL TIBIA : 5
- BL COXA : 13
- BL FEMUR : 15
- BL TIBIA : 17
- CL COXA : 7
- CL FEMUR : 9
- CL TIBIA : 11

- AR COXA : 2
- AR FEMUR : 4
- AR TIBIA : 6
- BR COXA : 14
- BR FEMUR : 16
- BR TIBIA : 18
- CR COXA : 8
- CR FEMUR : 10
- CR TIBIA : 12

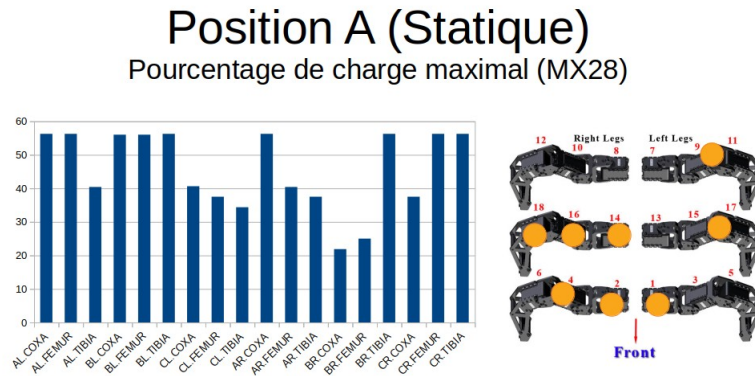


FIGURE 5.31 – Pourcentage de sollicitation d’effort sur les servomoteurs des différents segments en position statique (A)

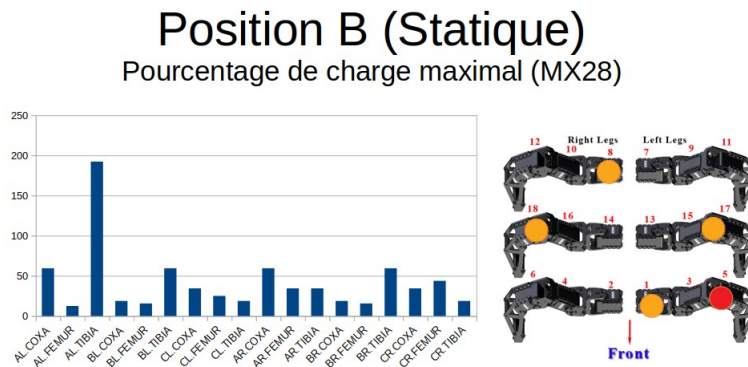


FIGURE 5.32 – Pourcentage de sollicitation d’effort sur les servomoteurs des différents segments en position statique (B)

Les efforts sont plus importants lorsque le robot est placé en position C, avec une garde au sol de 25 cm et moins important lorsque le robot est position B, avec une garde au sol de

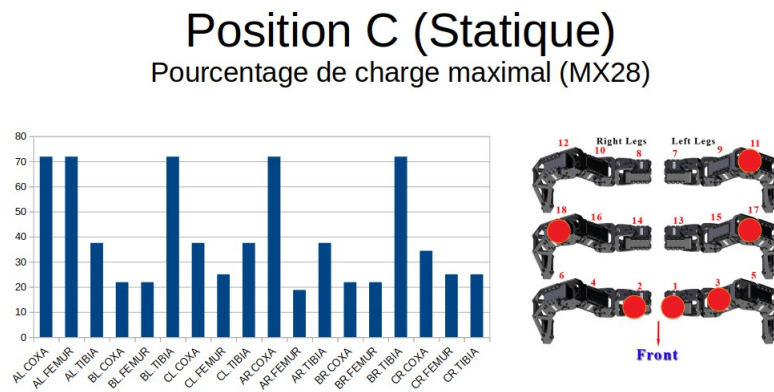


FIGURE 5.33 – Pourcentage de sollicitation d’effort sur les servomoteurs des différents segments en position statique (C)

15 cm. L’expérimentation a montré un déséquilibre du centre de gravité sur les 3 positions. La charge est principalement répartie vers l’avant. De plus, les servomoteurs subissent des efforts statiques de 50 % de la charge maximale admissible pour les servomoteurs. L’expérimentation a démontré que le centre de gravité sur la configuration actuelle est à repositionner. En effet, la charge vers l’avant est due aux équipements positionnés à cet endroit (caméra stéréoscopique et caméra VPU). De plus, les servomoteurs peuvent potentiellement être sous dimensionnés et atteindre leurs limites dans un environnement dynamique.

5.2.2 Résultats en état dynamique

Comme nous l’avons vu précédemment, des efforts trop importants sur les servomoteurs entraînent un arrêt de leur fonctionnement. Cette problématique entraîne des difficultés supplémentaires pour la locomotion. Dans cette partie, nous allons décrire les résultats des différents mouvements et leur impact sur les servomoteurs.

Essais dynamiques

Les essais dynamiques visent à analyser d’éventuelles surcharges de couple sur les servomoteurs en fonction des démarches. Ce test nous permet aussi de vérifier les défauts de parallélisme, de réglage de centre de masse et les efforts de bras de levier sur les pattes. Les essais sont réalisés sur un sol plat de type carrelage (coefficient de friction entre 0,50 et 1 μ k) en commande manuelle. Le test est réalisé avec une consigne de vitesse à 100

% vers les différentes possibilités de déplacement pendant 5 secondes dans une démarche tripode, qui est la plus critique, car 3 pattes assurent la stabilité comparée à une démarche de faucheur (succession de 2 pattes). Les mouvements testés sont la marche avant, arrière, la rotation gauche et droite, et les mouvements latéraux (gauche et droite).

Les objectifs sont d'évaluer les efforts sur des mouvements simples, repérer / évaluer les déplacements critiques et analyser les efforts des servomoteurs.

Nous souhaitons dans cet essai repérer les articulations sollicitées, les défauts d'équilibrage du centre de gravité. Les efforts ne doivent pas dépasser 100 % des couples maximums admissibles par les servomoteurs dynamixels MX28T (équipés sur ArIA Inspector), figure 5.34. Le couple de décrochage est de 2.3N.m (à 11.1V de tension et 1.3A de courant), 2.5N.m (à 12V, 1.4A), 3.1N.m (à 14.8V, 1.7A). Les essais sont réalisés avec une alimentation stabilisée de 12 V.



FIGURE 5.34 – Les servomoteurs Dynamixel MX28AT (Robotis) utilisés sur ArIA Inspector.

Résultats de marche avant et arrière Les résultats des essais de charge sur les servomoteurs sont illustré (Figure 5.35) pour la marche avant et (Figure 5.36) pour la marche arrière.

Les résultats sur la marche avant (Figure 5.35) montrent une plus grande charge sur les tibias, et plus important sur le côté droit (Milieu BR et arrière CR). Les fémurs du côté droit sont aussi touchés pour cet effet. Les coxas sont moins touchés par ce phénomène.

Les résultats sur la marche arrière (Figure 5.36) montrent une plus grande charge sur les tibias particulièrement du côté droit (avant AR et milieu BR). Les coxas sont aussi moins sollicités par les efforts.

Résultats des essais de charge sur les servomoteurs en marche avant

	CHARGE PATTE AVANT GAUCHE (AL)			CHARGE PATTE AVANT DROITE (AR)		
	AL.COXA	AL.FEMUR	AL.TIBIA	AR.COXA	AR.FEMUR	AR.TIBIA
MAXIMUM	0,0732	0,0674	0,2813	0,0869	0,2344	0,2344
MINIMUM	0,0000	-0,0732	-0,2197	-0,0752	-0,2861	-0,1582
MEDIANE	0,0024	-0,0121	0,0624	0,0210	-0,0968	0,0914
	CHARGE PATTE MILIEU GAUCHE (BL)			CHARGE PATTE MILIEU DROITE		
	BL.COXA	BL.FEMUR	BL.TIBIA	BR.COXA	BR.FEMUR	BR.TIBIA
MAXIMUM	0,2393	0,0762	0,2266	0,0723	0,3174	0,5469
MINIMUM	-0,2939	-0,2266	-0,2002	-0,3877	-0,2344	-0,2627
MEDIANE	-0,1304	-0,0054	0,0761	-0,0503	-0,0708	0,2173
	CHARGE PATTE ARRIERE GAUCHE (CL)			CHARGE PATTE ARRIERE DROITE		
	CL.COXA	CL.FEMUR	CL.TIBIA	CR.COXA	CR.FEMUR	CR.TIBIA
MAXIMUM	0,1719	0,2656	0,3799	0,0703	0,3281	0,4453
MINIMUM	-0,2012	-0,3438	0,0508	-0,2813	-0,2646	-0,1875
MEDIANE	-0,0488	-0,0540	0,1998	-0,0497	0,1020	0,1341

FIGURE 5.35 – Sollicitation d’effort sur les servomoteurs des différents segments en marche avant en daN.m.

Résultats des essais de charge sur les servomoteurs en marche arrière

	CHARGE PATTE AVANT GAUCHE (AL)			CHARGE PATTE AVANT DROITE (AR)		
	AL.COXA	AL.FEMUR	AL.TIBIA	AR.COXA	AR.FEMUR	AR.TIBIA
MAXIMUM	0,2285	0,0635	0,2891	0,2891	0,2412	0,4912
MINIMUM	-0,2813	-0,2578	-0,2422	-0,1484	-0,3672	-0,2842
MEDIANE	-0,0008	-0,0568	0,0942	0,0488	-0,1253	0,2050
	CHARGE PATTE MILIEU GAUCHE (BL)			CHARGE PATTE MILIEU DROITE		
	BL.COXA	BL.FEMUR	BL.TIBIA	BR.COXA	BR.FEMUR	BR.TIBIA
MAXIMUM	0,1445	0,1230	0,2783	0,1191	0,2412	0,4004
MINIMUM	-0,5195	-0,1768	-0,2393	-0,1494	-0,1875	-0,1982
MEDIANE	-0,0755	-0,0410	0,0979	0,0028	-0,0941	0,2159
	CHARGE PATTE ARRIERE GAUCHE (CL)			CHARGE PATTE ARRIERE DROITE		
	CL.COXA	CL.FEMUR	CL.TIBIA	CR.COXA	CR.FEMUR	CR.TIBIA
MAXIMUM	0,1270	0,2168	0,1533	0,2031	0,2383	0,2979
MINIMUM	-0,4141	-0,1641	-0,0645	-0,1436	-0,2246	-0,1172
MEDIANE	-0,1567	-0,0404	0,0542	0,0329	0,0536	0,0589

FIGURE 5.36 – Pourcentage de sollicitation d’effort sur les servomoteurs des différents segments en marche arrière en daN.m.

Résultats en rotation horaire et antihoraire. Les résultats de la rotation gauche et droite sont présenté respectivement Figure 5.37 et Figure 5.38.

Les résultats de la rotation gauche (Figure 5.37) montre une plus grande charge sur les tibias, et plus important sur le côté gauche (Milieu BL et arrière CL). Les fémurs et les coxis du côté droit sont aussi touchés pour cet effet.

Les résultats de la rotation droite (Figure 5.38) montrent une plus grande charge sur les tibias, et plus important sur le côté droit (Milieu BR et arrière CR). Les fémurs du côté droit sont aussi touchés pour cet effet. Les coxas sont moins touchés par ce phénomène.

Résultats des essais de charge sur les servomoteurs en rotation gauche (sens antihoraire)

	CHARGE PATTE AVANT GAUCHE (AL)			CHARGE PATTE AVANT DROITE (AR)		
	AL.COXA	AL.FEMUR	AL.TIBIA	AR.COXA	AR.FEMUR	AR.TIBIA
MAXIMUM	0,0000	0,1055	0,2578	0,2852	0,2100	0,2236
MINIMUM	-0,2969	-0,0547	-0,2148	-0,1035	-0,2266	-0,2129
MEDIANE	-0,0093	-0,0050	0,1000	0,0654	-0,0855	0,0840
ECART MOYEN MX28	-0,2549	-0,2530	-0,1631	-0,2093	-0,3229	-0,1908
	CHARGE PATTE MILIEU GAUCHE (BL)			CHARGE PATTE MILIEU DROITE (BR)		
	BL.COXA	BL.FEMUR	BL.TIBIA	BR.COXA	BR.FEMUR	BR.TIBIA
MAXIMUM	0,4297	0,0566	0,2266	0,1602	0,2393	0,3828
MINIMUM	-0,2998	-0,0674	-0,3037	-0,2109	-0,1484	0,0049
MEDIANE	-0,0868	0,0084	0,0934	-0,0010	-0,0815	0,2944
ECART MOYEN MX28	-0,3083	-0,2447	-0,1842	-0,2507	-0,3088	-0,0184
	CHARGE PATTE ARRIERE GAUCHE (CL)			CHARGE PATTE ARRIERE DROITE (CR)		
	CL.COXA	CL.FEMUR	CL.TIBIA	CR.COXA	CR.FEMUR	CR.TIBIA
MAXIMUM	0,1221	0,2344	0,3984	0,1152	0,2344	0,3828
MINIMUM	-0,3828	-0,2266	0,1689	-0,1484	-0,2510	-0,1465
MEDIANE	-0,2160	-0,0587	0,2550	-0,0003	0,0835	0,1483

FIGURE 5.37 – Sollicitation d’effort sur les servomoteurs des différents segments en rotation gauche en daN.m.

Résultats des essais de charge sur les servomoteurs en rotation droite (sens horaire)

	CHARGE PATTE AVANT GAUCHE (AL)			CHARGE PATTE AVANT DROITE (AR)		
	AL.COXA	AL.FEMUR	AL.TIBIA	AR.COXA	AR.FEMUR	AR.TIBIA
MAXIMUM	0,0391	0,0664	0,2295	0,0811	0,2598	0,3906
MINIMUM	-0,2891	-0,2373	-0,1631	-0,1172	-0,2813	-0,1475
MEDIANE	-0,0074	-0,0690	0,0925	0,0054	-0,1160	0,1944
	CHARGE PATTE MILIEU GAUCHE (BL)			CHARGE PATTE MILIEU DROITE (BR)		
	BL.COXA	BL.FEMUR	BL.TIBIA	BR.COXA	BR.FEMUR	BR.TIBIA
MAXIMUM	0,1719	0,0898	0,1953	0,0986	0,2441	0,5156
MINIMUM	-0,3672	-0,2188	-0,2422	-0,3242	-0,2969	-0,1377
MEDIANE	-0,1767	-0,0257	0,0813	-0,0328	-0,0974	0,3487
	CHARGE PATTE ARRIERE GAUCHE (CL)			CHARGE PATTE ARRIERE DROITE (CR)		
	CL.COXA	CL.FEMUR	CL.TIBIA	CR.COXA	CR.FEMUR	CR.TIBIA
MAXIMUM	0,2002	0,2334	0,1719	0,2002	0,2334	0,1719
MINIMUM	-0,2363	-0,1934	-0,0371	-0,2363	-0,1934	-0,0371
MEDIANE	-0,0907	-0,0509	0,1025	-0,0907	-0,0509	0,1025

FIGURE 5.38 – Sollicitation d’effort sur les servomoteurs des différents segments en rotation droite en daN.m.

Résultats en translation gauche et droite. Les résultats de la rotation gauche et droite sont présentés respectivement Figure 5.39 et Figure 5.40.

Les résultats de la translation à gauche (Figure 5.39) montrent une plus grande charge sur les tibias, et plus important sur le côté gauche (Milieu BL et arrière CL). Les fémurs et coxis du côté droit sont aussi touchés pour cet effet.

Les résultats de la translation à droite (Figure 5.37) montrent une plus grande charge sur les tibias, et plus important sur le côté droit (Milieu BR et arrière CR). Les fémurs et coxis du côté droit sont aussi touchés pour cet effet.

Résultats des essais de charge sur les servomoteurs en translation sur la gauche

	CHARGE PATTE AVANT GAUCHE (AL)			CHARGE PATTE AVANT DROITE (AR)		
	AL.COXA	AL.FEMUR	AL.TIBIA	AR.COXA	AR.FEMUR	AR.TIBIA
MAXIMUM	0,0156	0,2148	0,3750	0,2813	0,2520	0,2666
MINIMUM	-0,7500	-0,0879	-0,2578	-0,0771	-0,3574	-0,2842
MEDIANE	-0,0290	0,0395	0,1853	0,0454	-0,0881	-0,0123
	CHARGE PATTE MILIEU GAUCHE (BL)			CHARGE PATTE MILIEU DROITE		
	BL.COXA	BL.FEMUR	BL.TIBIA	BR.COXA	BR.FEMUR	BR.TIBIA
MAXIMUM	0,3516	0,0566	0,5859	0,0625	0,2461	0,8076
MINIMUM	-0,0703	-0,1328	-0,2168	-0,1094	-0,4355	-0,1914
MEDIANE	0,0631	-0,0309	0,2650	-0,0266	-0,1937	0,2792
	CHARGE PATTE ARRIERE GAUCHE (CL)			CHARGE PATTE ARRIERE DROITE		
	CL.COXA	CL.FEMUR	CL.TIBIA	CR.COXA	CR.FEMUR	CR.TIBIA
MAXIMUM	0,0898	0,0898	0,4111	0,0869	0,5732	0,3594
MINIMUM	-0,6055	-0,6055	-0,2939	-0,2539	-0,4121	-0,1602
MEDIANE	-0,1735	-0,1735	-0,1160	-0,0615	0,1326	0,1466

FIGURE 5.39 – Sollicitation d’effort sur les servomoteurs des différents segments en translation gauche en daN.m.

Résultats des essais de charge sur les servomoteurs en translation sur la droite

	CHARGE PATTE AVANT GAUCHE (AL)			CHARGE PATTE AVANT DROITE (AR)		
	AL.COXA	AL.FEMUR	AL.TIBIA	AR.COXA	AR.FEMUR	AR.TIBIA
MAXIMUM	0,0215	0,5732	0,3320	0,1104	0,2783	0,1016
MINIMUM	-0,0029	-0,4121	-0,2656	-0,1172	-0,3281	-0,0918
MEDIANE	0,0006	0,1326	0,0841	0,0120	-0,1322	0,0215
ECART MOYEN MX28	-0,2497	-0,1199	-0,2004	-0,2447	-0,3302	-0,2412
	CHARGE PATTE MILIEU GAUCHE (BL)			CHARGE PATTE MILIEU DROITE		
	BL.COXA	BL.FEMUR	BL.TIBIA	BR.COXA	BR.FEMUR	BR.TIBIA
MAXIMUM	0,3271	0,0703	0,3691	0,2500	0,9502	0,5626
MINIMUM	-0,2510	-0,0469	-0,2578	-0,4922	-0,4766	-0,4634
MEDIANE	-0,0150	0,0094	0,1239	-0,1880	0,4733	0,3521
ECART MOYEN MX28	-0,2596	-0,2460	-0,1911	-0,3455	0,0681	0,1012
	CHARGE PATTE ARRIERE GAUCHE (CL)			CHARGE PATTE ARRIERE DROITE		
	CL.COXA	CL.FEMUR	CL.TIBIA	CR.COXA	CR.FEMUR	CR.TIBIA
MAXIMUM	0,1689	0,3193	0,2646	0,1660	0,2813	0,3555
MINIMUM	-0,1836	-0,3750	-0,1035	-0,0938	-0,3467	-0,0898
MEDIANE	-0,0076	-0,0782	0,1198	0,0149	0,0968	0,1313

FIGURE 5.40 – Pourcentage de sollicitation d’effort sur les servomoteurs des différents segments en translation droite.

5.2.3 Expérimentation et résultats du franchissement d’obstacles

Expérimentation du franchissement d’obstacles

Les essais de franchissement d’obstacles en simulation conservent les mêmes dispositions. Un surbau artificiel de 16 cm a été créé pour les essais de franchissement. Le franchissement d’obstacle se déroule en environnement Gazebo avec une modélisation d’une porte avec un surbau à partir des données du plan. Ce type d’obstacle est très récurrent dans le cas d’utilisation n°2, dans un environnement naval. L’opération est réalisée en commande manuelle. Cette simulation a pour but de préparer les réglages pour un essai client et de démontrer la faisabilité de traverser un couloir de frégate avec ces mêmes obstacles.

Test de franchissement

Les tests de franchissement d'obstacle sont essentiels pour déterminer la capacité maximale de franchissement. Les essais ont été réalisés en simulation sous Gazebo (Figure 5.41) et en essais réels (Figure 5.42). Les essais ont été réalisés avec succès dans le cas réel (Figure 5.41) en utilisant la version prototypée de ArIA Inspector. Un surbait artificiel a été créé pour atteindre 16 cm en taille d'obstacle.

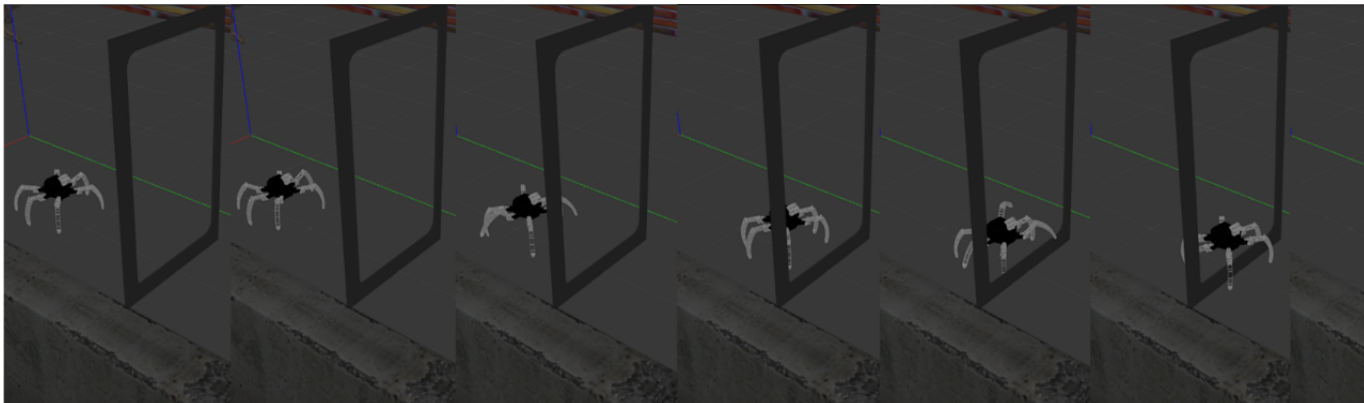


FIGURE 5.41 – Franchissement d'un surbait de 16 cm au sol en simulation sous Gazebo.

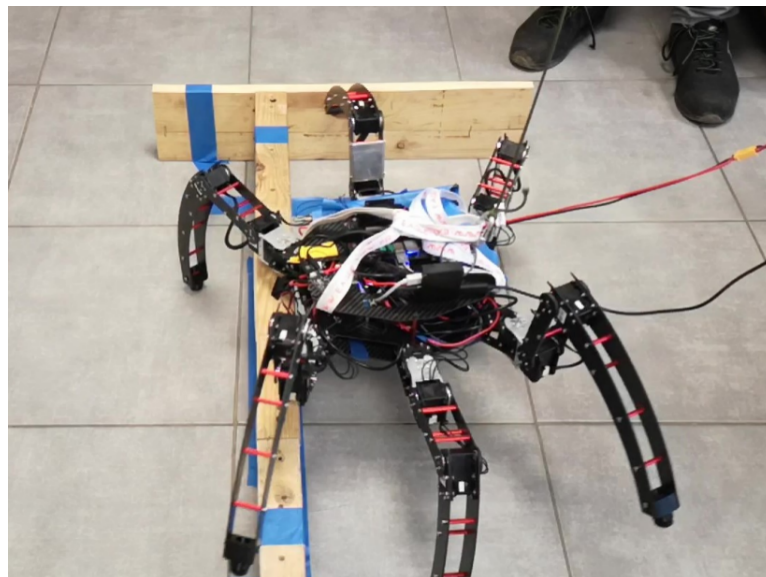


FIGURE 5.42 – Franchissement d'un surbait de 16 cm au sol en situation réelle.

Les essais ont été réalisés (Figure 5.42) en utilisant la version prototypée de ArIA Inspector. La vidéo des essais est disponible sur le lien suivant <https://www.youtube>.

[com/watch?v=LV7k4Cr7Ah0](https://www.youtube.com/watch?v=LV7k4Cr7Ah0).

5.2.4 Conclusion

Les efforts mécaniques engendrés lors de la démarche sont plus importants que les efforts statiques. Les essais démontrent que les servomoteurs sont sous-dimensionnés et qu'une capacité supérieure serait nécessaire. Durant les essais des translations gauches et droites, les servomoteurs impactés par cette sollicitation ont été mis en défaut. Ces essais concernent la démarche tripode, considérée comme la plus critique. Le design de ArIA Inspector a été conçu pour le franchissement d'obstacle et implique une longueur de tibia et fémur optimisant la portée des pattes. Cette longueur entraîne un effet de bras de levier, augmentant le couple exercé aux extrémités des liaisons mécaniques. Un équilibrage sur les fémurs et tibias permettrait de réduire ces efforts. Une augmentation des possibilités de charges des servomoteurs est à prévoir pour une prochaine version.

5.3 Présentation de scénarios représentatifs

Dans cette section, nous abordons un scénario de simulation chimique dans un environnement complexe (labyrinthe) en représentant le cas d'utilisation 2 (environnement de carrière souterraine pour la complexité des couloirs) et un essai représentant le cas d'utilisation 1 (environnement industriel et naval).

5.3.1 Simulation chimique : Labyrinthe

Dans cette partie, nous présentons les essais d'asservissement chimique dans un labyrinthe. L'essai prend environ 29 minutes dans un environnement labyrinthe à compléter, ce qui plus lent par rapport à une plateforme mobile (environ 5 minutes).

Le lien vers la vidéo : Méthode "Grid" <https://youtu.be/cmi3geRgZbg> et méthode "Surge-Cast" https://www.youtube.com/watch?v=iKUH13nT_o.

Les résultats montrent la trajectoire (Figure 5.43) en cm, la position de comparaison entre la cible et le robot et l'erreur relative sur les axes X et Y du robot à l'intérieur du labyrinthe.

Ces résultats montrent que l'erreur relative est de 180 cm sur l'axe Y et de 60 cm sur l'axe X .

L'asservissement chimique dans un labyrinthe a été réalisé en utilisant un robot hexapode. Durant l'asservissement chimique, nous trouvons des problématiques de temps. Le robot hexapode à une locomotion moins rapide qu'un robot à roues. Nous retrouvons les erreurs de trajectoire vues précédemment. Ces oscillations de navigation répétées surviennent principalement quand le robot est en sortie des virages du labyrinthe.

La difficulté de cet environnement résidait dans sa complexité géométrique. Les trajectoires associées sont plus complexes à appréhender que dans l'environnement précédant. Les erreurs de trajectoires sont en partie dues au réglage des cartes de coût. Le réglage reste délicat pour un robot hexapode dans un milieu contraint. Une carte de coût trop large, par exemple, ne permettra pas au robot d'accéder à des zones contraintes, mais évitera largement les obstacles. Au contraire, une carte de coût trop étroite, permettra au robot de franchir les milieux restreints, mais les pattes pourront potentiellement rencontrer des obstacles et s'abîmer en situation réelle.

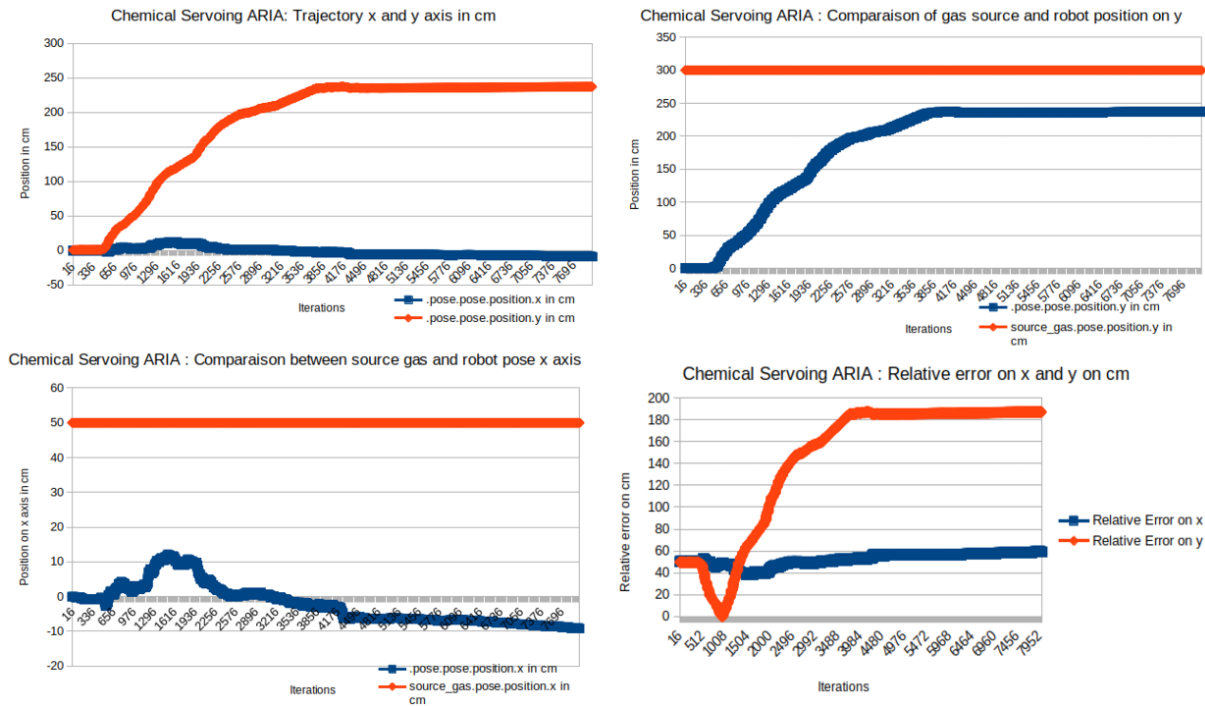


FIGURE 5.43 – Asservissement chimique : environnement du labyrinthe.

5.3.2 Essai réalisé à l'entreprise Naval Group

Une démonstration a été réalisée pour le franchissement d'obstacle de surbaut dans une course de navire. L'objectif est de comparer l'évolution de ArIA Inspector, une solution marchante et une solution à chenille, FurIA.

Les modèles testés lors de cette démonstration sont les modèles ArIA Inspector, présenté 5.45 et FurIA Track XXL, représenté par la première version (FurIA Track) 5.44. Dans cette démonstration, deux architectures de robots (marchante et roulante) sont testés dans un environnement naval.



FIGURE 5.44 – Furia Track, le robot à chenille développé par Pilgrim Technology



FIGURE 5.45 – ArIA INSPECTOR, le robot hexapode développé par Pilgrim Technology

Les caractéristiques du robot ArIA INSPECTOR et FurIA Track XXL sont présentées dans le tableau 5.3.2.

Caractéristiques	Données ArIA
Poids	4,4 kg
Dimensions	40cm, 40cm, 46cm (diagonale)
Garde au sol	35cm
Pente	35 degrés.
Franchissement	20 cm (maximum).
Vitesse	0,10m/s
Emport	0,5 kg
Mode de pilotage	Manette XBox360.
Communication	WiFi longue portée 2.4 GHz.
Cas d'usage	Inspection, Surveillance, Reconnaissance, Militaire.

Caractéristiques	Données FurIA Track XXL
Poids	35 kg
Dimensions	Longueur 85 cm largeur 51 cm hauteur 34 cm
Franchissement	18 cm
Pente	40 degrés
Vitesse	3,6 km/h
Roulis	35 degrés
Emport	20 kg
Mode de pilotage	Radio-commande
Communication	Radiocommunication 2,4 GHz.
Cas d'usage	Inspection, Surveillance, Militaire.

Les deux robots ArIA 5.3.2 et FurIA 5.3.2 ont été télé opérés chacun leur tour dans l'environnement de démonstration client. Le parcours comportait plusieurs surbauts de 20 cm, 16 cm et 10 cm. ArIA Inspector a réussi les franchissements d'obstacle de 16 et 10 cm. En revanche, les surbauts de 20 cm n'ont pas été testés. L'expérimentation montre un robot marcheur à plus de facilité à franchir les obstacles pour un poids de robot ne dépassant pas les 5 kg. En revanche, la vitesse de sa locomotion présente un frein, actuellement, pour les interventions dites "rapide".

Les tests concernant ArIA Inspector ont été focalisés sur le franchissement d'obstacles. Le passage d'un surbaut de 16 cm a été réalisé en 13 secondes et le franchissement d'un double surbaut de 16 cm a duré 1 minute. L'appréhension des angles à 90 degrés des couloirs a aussi été validé durant l'expérimentation. Le temps pour réaliser un tour complet (2π radians) est de 12 secondes. La vitesse du robot est un axe d'amélioration, car les courives représentent une distance supérieure à 150 mètres. La vitesse du robot FurIA est un point fort sur cette architecture de robot. Les longues courives peuvent être, donc, être inspectées plus rapidement. Le franchissement d'un surbaut de 16 cm peut être effectué en 3 secondes, malgré la stratégie du passage en force dû à la locomotion. L'hexapode possède 2 modes de pilotage (réglage manuel) permettant le passage de surbaut et déplacement dans une courive (milieu contraint). Le passage en milieu contraint a en conséquence été validé pendant l'expérimentation. Le déplacement est lent, car encore en phase prototype. Mais l'objectif pour ARIA est de développer l'aspect logiciel en ajoutant d'autres fonctionnalités (la détection et franchissement automatique d'obstacle) et améliorer sa vitesse. Le robot a la possibilité de diriger les pattes en mode 3 par 3 ou 2 par 2.

5.4 Conclusion de fin de chapitre

Dans cette étude, nous présentons deux méthodes distinctes pour réaliser un asservissement visuel (méthode par QRcode) ou chimique (méthode par recherche de grille). Au cours des différentes expérimentations, nous avons réalisé l'importance d'adapter les paramètres du contrôleur et qui peuvent influencer l'efficacité et la fiabilité de la méthode d'asservissement. L'approche par réseaux de neurones est orientée sur une méthode supervisée, nécessitant une grande quantité d'images à labelliser pour permettre l'entraînement. La base de données présentée peut-être davantage plus dense en créant un processus d'acquisition et de traitement des images de défauts lors d'inspections industrielles. La valeur ajoutée de l'approche est de pouvoir transférer les connaissances des défauts dans une méthode de détection automatisée. Cependant, la labellisation est, aujourd'hui, une opération manuelle nécessitant une personne avec des connaissances sur les défauts industriels. Une approche automatisée de labellisation permettrait d'obtenir un processus de production d'inférence qui serait pertinente par rapport aux défauts rencontrés.

Dans notre approche d'asservissement chimique, nous proposons l'utilisation de capteurs oxyde de métal. Cette approche diffère de l'asservissement visuel utilisé et nous avons donc deux algorithmes différents. L'utilisation d'une caméra chimique permettrait d'utiliser une méthode jacobienne, utilisé lors de l'asservissement visuel, pour traiter les POI chimiques. De ce fait, une seule méthode pourrait traiter les percepts visuels et chimiques. Les tests conduits ont cependant permis de démontrer la faisabilité des différents procédés. Les tests d'odométrie et de navigation ont été davantage des tests de réglages. Un échantillonnage plus grand permettrait de définir précisément l'efficacité des procédés. L'IMU, testé lors de l'odométrie, produisait beaucoup de bruit en simulation, malgré les filtres et réglages. Néanmoins, une IMU conjuguée à une méthode de correspondance de scans laser (en fusion de capteur), peut permettre d'avoir de meilleurs résultats en simulation ou en réel. Les performances de l'entraînement de YOLO v5 sur la base de données proposée sont validés par les différentes métriques et répondent aux besoins de Pilgrim Technology.

De plus, une approche par apprentissage par renforcement contribuerait à éviter les directions s'éloignant de la cible ou de la source du gaz et améliorerait le comportement global d'asservissement (temps, trajectoire, etc.). L'architecture de ce robot permet aussi de franchir des obstacles et des essais réels ont été réalisés chez un client industriel, Naval Group. Par ces essais, nous souhaitons donc qualifier le robot pour les différents environnements liés aux cas d'étude et aux applicatifs d'inspections industrielles.

Le déploiement et les différents essais réalisés ont nécessité des multiples manipulations pendant les différentes phases de développement. Par exemple, la réalisation d'un stand a été nécessaire pour paramétrer la configuration des pattes et un harnais de démarche pour les premiers pas du prototype et les essais de franchissement d'obstacles de 16 cm. Il serait nécessaire de pousser les essais vers des cas extrêmes d'environnements contraints et dégradés en termes de géométrie, d'obstacles et de configuration. Il est donc encore tôt pour déterminer les limites d'exploration concernant le prototype développé. Néanmoins, le robot a démontré lors du cas client, que les capacités du robot actuel permettaient de réaliser l'exploration totale du trajet proposé par le client. Le franchissement d'obstacles a été démontré en simulation et en réel lors d'un essai à l'intérieur d'une frégate.

Nous pouvons recommander, autre que la diversité des environnements lors des essais, d'accroître le nombre d'itérations lors des essais afin de constituer des statistiques davantage robustes et représentatives d'un usage récurrent du prototype. Des essais supplémentaires pourraient indiquer l'énergie consommée, les couples lors d'un test de franchissement ou d'asservissement. Des tests pourraient être faits pour évaluer le comportement du robot lors d'un asservissement avec obstacles. Des essais supplémentaires seraient à prévoir pour déterminer le couple engagé sur les différents segments du robot et affiner la stratégie de franchissement d'obstacle.

CONCLUSIONS : CONTRIBUTIONS ET PERSPECTIVES

Dans ce chapitre, nous proposons d'aborder les contributions de cette étude et de proposer les perspectives et les ouvertures pour les futurs travaux de recherche et de développement.

6.1 Contributions

Dans cette étude, les contributions sont multiples. 1) Le postulat est de proposer une solution de cotation et d'analyse d'environnement servant de données quantifiables pour concevoir le robot. 2) Nous proposons une architecture de robot hexapode permettant de surmonter les surbauts de hauteur prédéfinie. 3) Nous proposons une architecture de navigation permettant la navigation en espaces contraints. 4) Cette même navigation sert à l'intégration des points de navigation traités par les méthodes d'asservissement, visuel et chimique. 5) Les tests nous ont montrés que l'asservissement visuel et chimique était réalisable en milieu contraint. 6) Le robot hexapode a démontré, lors de l'essai client, qu'il possède une meilleure capacité de franchissement d'obstacle en étant moins bruyant et moins lourd que le robot à chenille.

Certains points sont à nuancer, le robot hexapode se déplace plus lentement en comparatif à d'autres types de robots. Néanmoins, le but final de cette preuve de concept est le domaine générique de l'inspection. Dans ce domaine, le but n'est pas d'aller vite, mais de récupérer (et traiter) un maximum d'informations par le biais des capteurs. Les prises de vues, possibilités de positionnement, l'agilité et la fluidité particulière de cette architecture d'hexapode peuvent contribuer à l'efficacité des algorithmes (stabilité) mais peuvent aussi ajouter du bruit (bruit mécanique, oscillation mécanique, variation de hauteur lors de la marche). Les recherches ont aussi mené vers une solution d'asservissement visuel et chimique en utilisant des méthodes existantes. La méthode d'asservissement

visuel par QRcode et la méthode d'asservissement chimique par la méthode "Grid" ont été démontrées précédemment. Les travaux de recherche menés sur la détection automatique de points d'intérêts industriels ont permis de réaliser une gamme de trois produits marchants (ArIA AX18, ArIA LUCAN, ArIA Inspector). Nous avons aussi employé une odométrie adaptée à l'architecture d'un robot hexapode à l'aide d'un Lidar 2D. Les essais ont démontré sa fiabilité, même si l'erreur d'orientation Z peut être encore améliorée. Une base de données de défauts industrielles a été constituée. Cette base de données labellisées nous a permis d'entraîner un réseau de neurones convolutif (Yolo5) afin de détecter automatiquement les défauts. Les poids synaptiques sont implémentables sur un système embarqué (Jetson Nvidia) ou sur une architecture Android pour une utilisation pédestre pour les inspecteurs.

6.2 Discussions

L'orientation de l'étude a été guidée conjointement par les besoins de l'entreprise en matière de prestation d'inspection et l'étude contextuelle de l'environnement des différents cas d'application. Si l'accent a été porté sur la résolution des différents cas, l'architecture robotique aura très certainement de meilleures facilités sur certaines configurations d'environnements (terrain difficile, sol glissant, franchissement d'obstacle, pentes) et à contrario, aura des difficultés particulières sur d'autres situations (descendre et monter des escaliers). Cependant, les possibilités de versatilités concernant les robots marchants sont démontrées naturellement par les espèces vivantes ayant ce type de configuration (les humains, chats, chiens, fourmis, araignées, par exemple). Dans cette partie, nous allons aborder les éléments concernant les axes d'améliorations de l'étude et le prototype développé. Ce travail de recherche propose une solution pour localiser et faire naviguer le robot dans un espace contraint, et ainsi le diriger automatiquement vers les percepts visuels et chimique. Cette solution a été démontrée en simulation. La théorie du Jacobien visuel pour les défauts industriels pourra être implémentée par la suite et l'ensemble des méthodes seront testées sur le robot réel. Les paquets ROS permettent de l'implémenter sur le robot physique, ArIA Inspector. La capacité d'autres capteurs, extéroceptifs et intéroceptifs, offrira de meilleures performances sur l'asservissement visuel ou autre. GADEN [176] a été, au cours du projet, a été, au cours du projet, a été restructuré sur le plan logiciel afin de faciliter son utilisation avec ROS (architecture des paquets et fichiers de lancement) en intégrant une configuration de l'environnement Gazebo. La perspective de ce travail est

d'ajouter de nouveaux algorithmes d'apprentissage pour la vision et poursuivre le développement de l'asservissement visuel sur des objets détectés par apprentissage automatique. Les réglages de la navigation devront être davantage optimisés pour avoir de meilleurs résultats sur les couloirs et virages contraints.

Le dimensionnement des pattes a été une décision complexe à réaliser. L'objectif initial était de développer un ratio de dimensionnement optimisant au mieux les capacités de locomotion, mais aussi de franchissement. Lors de l'étude, nous avons concentré l'effort sur un ratio permettant d'être moins énergivore sur les opérations de locomotion et franchissement d'obstacles. ArIA INSPECTOR a pu réaliser les essais de franchissement et de locomotion, mais plusieurs problématiques sont survenues lors des essais. Nous avons vu précédemment les problématiques liées à la conception des pattes. Dans cette partie, nous abordons les problèmes des conceptions du corps du robot. L'équilibre du centre de masse lors de l'intégration des différents équipements n'a pas été suffisamment optimisé et le centre de masse est décalé par rapport au prototype, avec davantage de poids vers l'avant, entraînant un déséquilibre mécanique et un effort parasite sur les différentes pattes. À l'origine, le système embarqué utilisé était une Jetson Nano de Nvidia. Si pour le fonctionnement du contrôleur, la charge calculatoire était suffisante, l'intégration des briques d'asservissement et reconnaissance visuel (YOLO v5), nécessitait un changement d'ordinateur embarqué. Le choix s'est dirigé vers une Jetson Xavier NX, permettant une plus grande capacité de calcul et une meilleure intégration pour les inférences. La première problématique est la possibilité pour le robot de basculer en arrière et de se renverser. Ce phénomène se produit lorsque une des pattes s'appuie sur la paroi de l'obstacle et que le robot est dans une posture de "contre-plongé" (avec un angle d'environ 45°). La seconde problématique est l'effet de bras de levier des pattes entraînant un effort supérieur à $2,5Nm$, capacité maximale des servomoteurs MX28T. La troisième problématique est liée aux pièces imprimés 3D (PLA) pour les différents segments des pattes. Lors de la démarche, les efforts engendrés détérioraient les pièces liant le fémur et le tibia, entraînant la rupture de ces pièces. De plus, des effets de torsion sur la liaison fémur-tibia ont été observés. Ces pièces ont été remplacées par des pièces usinées en aluminium pour corriger cette problématique. À l'origine, il a été prévu une tête où les capteurs peuvent être orientés vers la source du percept ou explorer plus facilement l'environnement. Cet aspect-là a été mis de côté par l'entreprise pour concentrer l'étude sur le robot et l'aspect capteur. Cependant, l'état actuel du robot ne permet pas d'inspecter 100% du volume, étant donné que l'inspection du plafond ou des surfaces hautes ne serait possible qu'en

orientant le corps du robot de sorte à orienter le flux de la caméra vers l'endroit souhaité. Cette opération nécessite donc de mobiliser 18 servomoteurs et sur un volume important, elle constitue un mouvement énergivore par rapport à la mobilisation de 2 servomoteurs pour la tête et une forte limitation opérationnelle dans le domaine de l'inspection industrielle.

La problématique de stabilité d'un robot hexapode peut entraîner une usure prématurée des servomoteurs, un basculement ou une chute du robot, particulièrement sur le franchissement d'obstacle. La stabilité des robots à pattes est divisée en deux catégories : la stabilité statique et la stabilité dynamique. Pour être considéré comme statiquement stable, le robot doit être stable pendant tout son cycle de marche, sans avoir besoin d'aucune force pour équilibrer le robot. Alors que le robot est statiquement stable, la projection verticale de son centre de masse (COM) est située à l'intérieur du polygone de support qui est formé entre les jambes qui sont en phase d'appui. Dans le cas où le centre de masse serait positionné en bordure ou à l'extérieur du polygone de support, le robot tombe à moins qu'il ne soit dynamiquement stable, c'est-à-dire que le robot soit équilibré en marchant en raison de l'inertie provoquée par le mouvement et est statiquement instable lorsqu'il s'arrête de bouger.

Les études de stabilité passent par les équations du mouvement (3.19). Les contraintes par rapport au sol doivent être suffisantes afin de produire uniquement des accélérations selon les directions et les orientations conformes au trajet désiré. Dans le cas contraire, la marche devient instable et il en résulte une possibilité de chute du marcheur.

La navigation du robot a été un sujet important et nécessaire pour la réalisation des tâches d'asservissement. À l'heure actuelle, le robot se déplace de manière non-holonyme. La problématique réside dans la configuration de la planification de trajectoire et du coût local de navigation (intrinsèque au robot). Plusieurs essais de configuration de planification de trajectoire ont été effectués pour permettre un mouvement holonyme. Des recherches plus approfondies sur des architectures de robots marchants utilisant ROS sont nécessaires pour débloquer ce point. Le mouvement holonyme représenterait un avantage en matière de locomotion et de comportement de recherche dû à davantage de libertés (déplacement sur l'axe x, sur l'axe y et les deux conjugués). Les procédés de navigation s'effectuent correctement, même si plusieurs essais sont nécessaires afin d'avoir un retour statistique de l'erreur et de réaliser des tests de fusion de capteurs. Un capteur laser (Lidar 2D) et un capteur de profondeur (Caméra stéréoscopique) permettrait d'avoir une meilleure précision odométrique et une cartographie plus complète (3D).

6.3 Perspectives de l'étude

Dans la simulation, les coordonnées des effecteurs sont traitées via la cinématique directe et inverse. Des réglages plus précis de la navigation, de la caméra, des capteurs, réduirait les erreurs potentielles et améliorera le comportement global du robot. L'intégration des fins d'effecteurs dans le procédé de navigation ("footprint", stance/stride phase) permettra de s'approcher au mieux du percept. Des modifications sur le contrôleur pour avoir des démarches qui peuvent évoluer durant l'asservissement lors d'obstacles par exemple. Le comportement de récupération pour la navigation et asservissement chimique doit être revue pour améliorer les résultats en cas d'absence de mesures soudaine du flux chimique (débit d'air et concentration). Les approches réalisées se focalise sur un effet de mesure et réaction. Le facteur vitesse étant plus faible qu'un robot à roue. L'apprentissage par renforcement pourrait limiter la perte du percept et valoriser un comportement de suivi du panache gazeux. Nous n'avons pas implémenté de capteurs de force, figure 6.1 pour la version d'ArIA Inspector sur les fins d'effecteurs. L'ajout d'une information de force sous les pieds de l'hexapode permettra d'avoir une autre source d'odométrie, améliorant la navigation et le système d'asservissement. En outre, les données de retour de force pourraient également ajouter une nouvelle stratégie de franchissement d'obstacle et seront essentielles pour monitorer davantage les efforts. Concernant les résultats d'asservissement, le robot atteint la cible, mais passe davantage de temps que pour un robot à roue.

Pour l'asservissement chimique, nous traitons facilement la source de gaz cible directe sans aucun obstacle. À l'intérieur d'un labyrinthe (29 minutes), l'efficacité diminue rapidement en raison de la navigation adoptée et doit être améliorée, notamment les paramètres des cartes de coût. Davantage d'essais sont nécessaires afin de perfectionner au mieux les réglages. Des essais complémentaires en réel, seront nécessaires afin de valider le procédé sur un cas réel d'étude. Concernant l'asservissement chimique sur le vrai robot, nous prévoyons d'utiliser de la fumée de glycol pour éviter les gaz dangereux et avoir une vue visuelle du panache. Pour l'instant, la navigation est réglée sur une configuration non-holonome, et nécessitera quelques capteurs (capteurs de force en fin d'effecteurs) et paramètres pour procéder à la navigation en mode holonome. L'empreinte de base n'est pas parfaitement configurée et devra s'ajuster au mieux entre le modèle théorique, le modèle URDF et le robot réel.

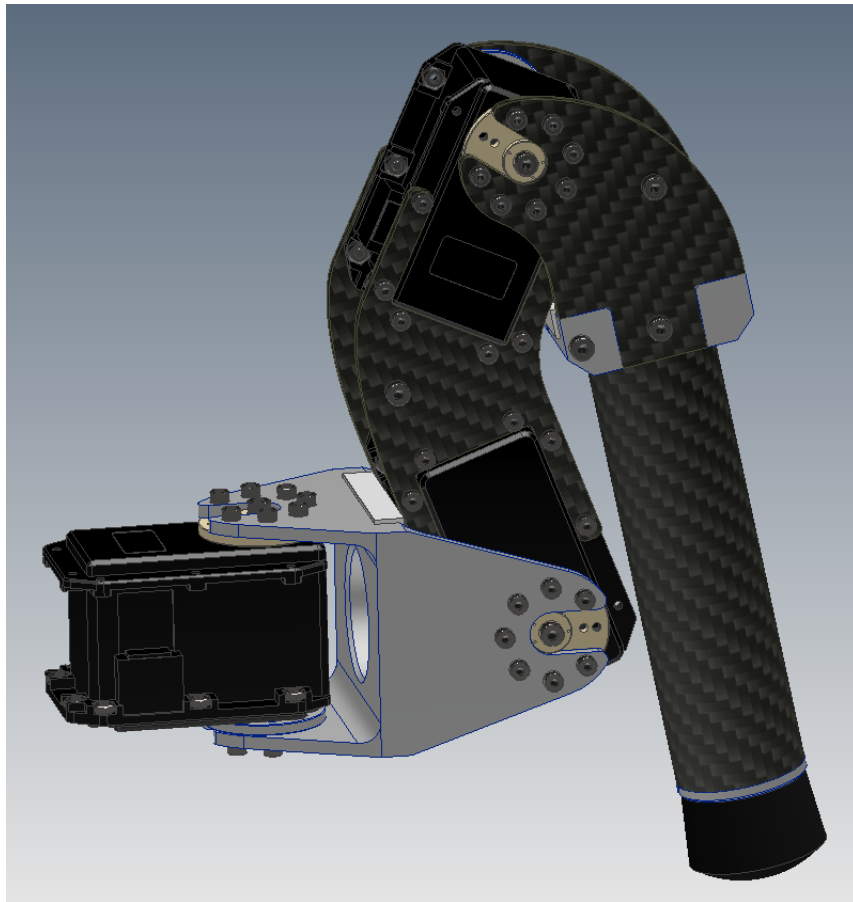


FIGURE 6.1 – Nouveau design de patte prenant en compte l'effort mécanique et l'intégration des servos MX106 et MX64

ANNEXES

7.1 ANNEXE A : Capteurs de gaz

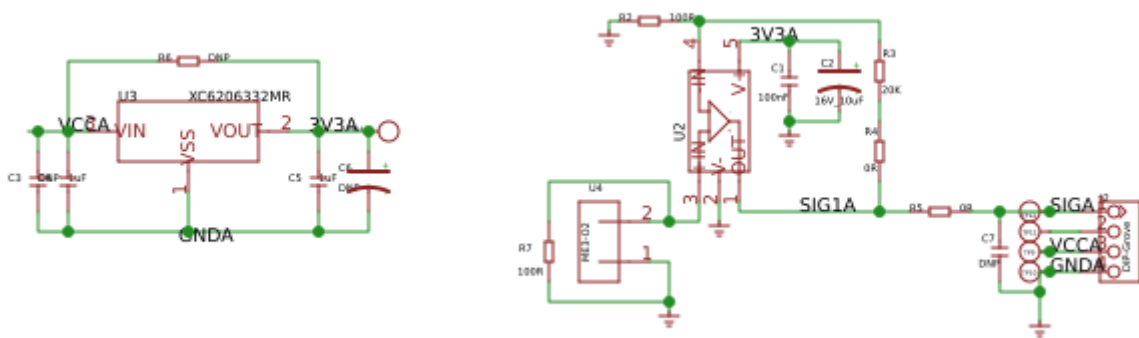


FIGURE 7.1 – Schéma électronique d'un capteur de dioxygène.

7.2 ANNEXE B : Logigramme de l'approche

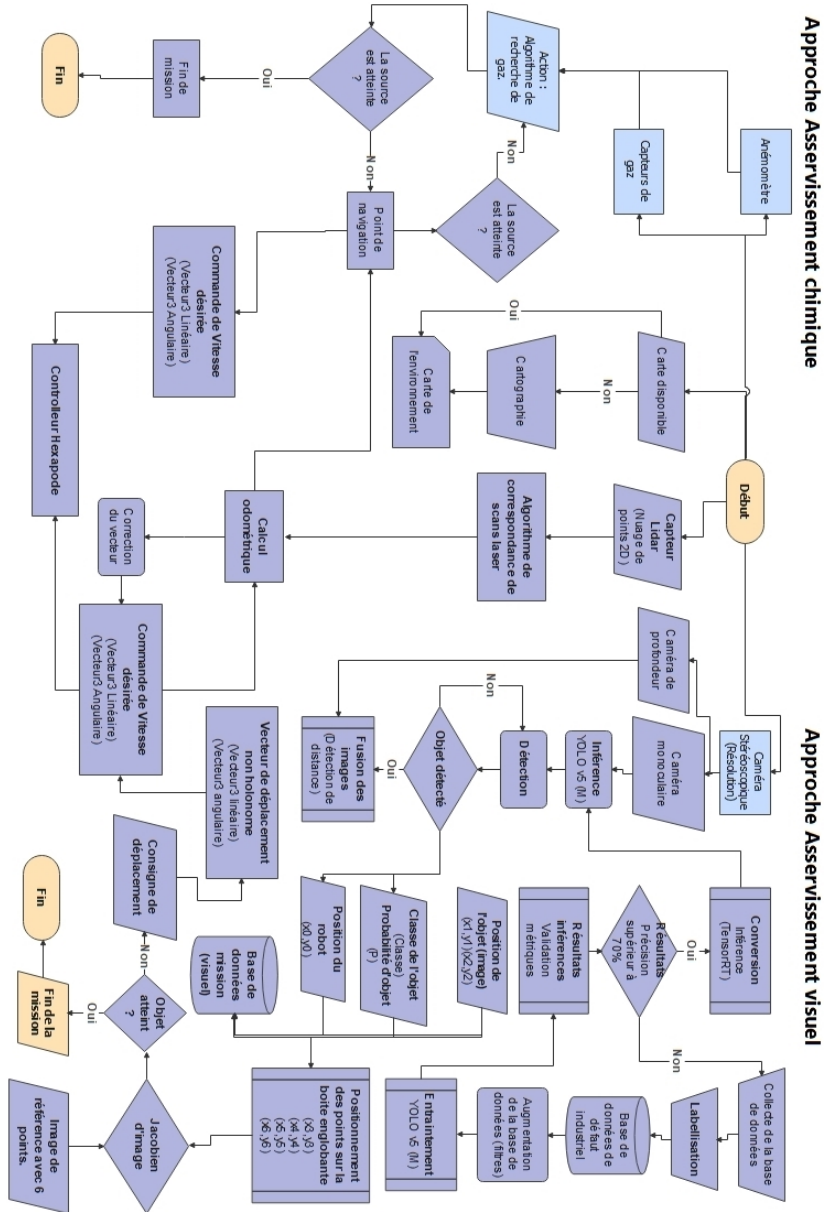


FIGURE 7.2 – Logigramme de l'approche de résolution pour un procédé d'asservissement visuel et chimique

7.3 ANNEXE C : Fichier de configuration (json) de la cinématique d'ArIA Inspector

```
# Model parameters
syropod_type: aria_inspector
leg_id:
[AR, BR, CR, CL, BL, AL] # leg_id_name (clockwise from front right leg)
joint_id:
[coxa, femur, tibia] # joint_id_name
link_id:
[base, coxa, femur, tibia] # link_id_name (always start with base)
leg_DOF:
{AR: 3, BR: 3, CR: 3, CL: 3, BL: 3, AL: 3}
# leg_id_name: degrees_of_freedom (num joints)

# Joint parameters:
AR_coxa_joint_parameters:
{offset: 0.0, min: -1.22, max: 1.22, packed: 0, unpacked: 0.0, max_vel: 5.0}
AR_femur_joint_parameters:
{offset: 0.0, min: -3.14, max: 1.57, packed: -1.57, unpacked: -0.79, max_vel: 5.0}
AR_tibia_joint_parameters:
{offset: 0.0, min: -2.5, max: 0, packed: 0, unpacked: -1.70, max_vel: 5.0}

BR_coxa_joint_parameters:
{offset: 0.0, min: -1.22, max: 1.22, packed: 0, unpacked: 0.0, max_vel: 5.0}
BR_femur_joint_parameters:
{offset: 0.0, min: -3.14, max: 1.57, packed: -1.57, unpacked: -0.79, max_vel: 5.0}
BR_tibia_joint_parameters:
{offset: 0.0, min: -2.5, max: 0, packed: 0, unpacked: -1.70, max_vel: 5.0}

CR_coxa_joint_parameters:
{offset: 0.0, min: -1.22, max: 1.22, packed: 0, unpacked: 0.0, max_vel: 5.0}
CR_femur_joint_parameters:
{offset: 0.0, min: -3.14, max: 1.57, packed: -1.57, unpacked: -0.79, max_vel: 5.0}
CR_tibia_joint_parameters:
```

```

{offset: 0.0, min: -2.5, max: 0, packed: 0, unpacked: -1.70, max_vel: 5.0}

CL_coxa_joint_parameters:
{offset: 0.0, min: -1.22, max: 1.22, packed: 0, unpacked: 0.0, max_vel: 5.0}
CL_femur_joint_parameters:
{offset: 0.0, min: -3.14, max: 1.57, packed: -1.57, unpacked: -0.79, max_vel: 5.0}
CL_tibia_joint_parameters:
{offset: 0.0, min: -2.5, max: 0, packed: 0, unpacked: -1.70, max_vel: 5.0}

BL_coxa_joint_parameters:
{offset: 0.0, min: -1.22, max: 1.22, packed: 0, unpacked: 0.0, max_vel: 5.0}
BL_femur_joint_parameters:
{offset: 0.0, min: -3.14, max: 1.57, packed: -1.57, unpacked: -0.79, max_vel: 5.0}
BL_tibia_joint_parameters:
{offset: 0.0, min: -2.5, max: 0, packed: 0, unpacked: -1.70, max_vel: 5.0}

AL_coxa_joint_parameters:
{offset: 0.0, min: -1.22, max: 1.22, packed: 0, unpacked: 0.0, max_vel: 5.0}
AL_femur_joint_parameters:
{offset: 0.0, min: -3.14, max: 1.57, packed: -1.57, unpacked: -0.79, max_vel: 5.0}
AL_tibia_joint_parameters:
{offset: 0.0, min: -2.5, max: 0, packed: 0, unpacked: -1.70, max_vel: 5.0}

# Link DH parameters:
# Ensure theta values are within joint limits

AR_base_link_parameters: {d: 0.036, theta: -0.54, r: 0.165505, alpha: 0.0}
AR_coxa_link_parameters: {d: 0.0156, theta: -0.245, r: 0.04468, alpha: 1.57}
AR_femur_link_parameters: {d: 0.0, theta: 1.57, r: 0.110, alpha: 0.0}
AR_tibia_link_parameters: {d: 0.0, theta: -0.51, r: 0.26603, alpha: 0.0}

BR_base_link_parameters: {d: 0.036, theta: -1.57, r: 0.146545, alpha: 0.0}
BR_coxa_link_parameters: {d: 0.0156, theta: 0.0, r: 0.04468, alpha: 1.57}
BR_femur_link_parameters: {d: 0.0, theta: 1.57, r: 0.110, alpha: 0.0}
BR_tibia_link_parameters: {d: 0.0, theta: -0.51, r: 0.26603, alpha: 0.0}

```

CR_base_link_parameters: {d: 0.036, theta: -2.6, r: 0.165505, alpha: 0.0}
CR_coxa_link_parameters: {d: 0.0156, theta: 0.245, r: 0.04468, alpha: 1.57}
CR_femur_link_parameters: {d: 0.0, theta: 1.57, r: 0.110, alpha: 0.0}
CR_tibia_link_parameters: {d: 0.0, theta: -0.51, r: 0.26603, alpha: 0.0}

AL_base_link_parameters: {d: 0.036, theta: 0.54, r: 0.165505, alpha: 0.0}
AL_coxa_link_parameters: {d: 0.0156, theta: 0.245, r: 0.04468, alpha: 1.57}
AL_femur_link_parameters: {d: 0.0, theta: 1.57, r: 0.110, alpha: 0.0}
AL_tibia_link_parameters: {d: 0.0, theta: -0.51, r: 0.26603, alpha: 0.0}

BL_base_link_parameters: {d: 0.036, theta: 1.57, r: 0.146545, alpha: 0.0}
BL_coxa_link_parameters: {d: 0.0156, theta: 0.0, r: 0.04468, alpha: 1.57}
BL_femur_link_parameters: {d: 0.0, theta: 1.57, r: 0.110, alpha: 0.0}
BL_tibia_link_parameters: {d: 0.0, theta: -0.51, r: 0.26603, alpha: 0.0}

CL_base_link_parameters: {d: 0.036, theta: 2.6, r: 0.165505, alpha: 0.0}
CL_coxa_link_parameters: {d: 0.0156, theta: -0.245, r: 0.04468, alpha: 1.57}
CL_femur_link_parameters: {d: 0.0, theta: 1.57, r: 0.110, alpha: 0.0}
CL_tibia_link_parameters: {d: 0.0, theta: -0.51, r: 0.26603, alpha: 0.0}

7.4 ANNEXE D : Réglage PID : Génération des paramètres (Pid.cfg)

```
#!/usr/bin/env python
PACKAGE = "pid"

from dynamic_reconfigure.parameter_generator_catkin import *

gen = ParameterGenerator()
K_scale_enum = gen.enum([ gen.const("scale_tenth", double_t, 0.1, "Scale by 0.1"),
gen.const("scale_unity", double_t, 1.0, "No scaling"),
gen.const("scale_ten", double_t, 10.0, "Scale by 10"),
gen.const("scale_hundred", double_t, 100.0, "Scale by 100")],
"Scale factor for K setting")

gen.add("Kp_scale", double_t, 0, "Kp scale", 10, 0.1, 100, edit_method = K_scale_enum)
gen.add("Kp", double_t, 0, "Kp", 0.1, -1, 1)
gen.add("Ki_scale", double_t, 0, "Ki scale", 10, 0.1, 100, edit_method = K_scale_enum)
gen.add("Ki", double_t, 0, "Ki", 0.1, -1, 1)
gen.add("Kd_scale", double_t, 0, "Kd scale", 10, 0.1, 100, edit_method = K_scale_enum)
gen.add("Kd", double_t, 0, "Kd", 0.1, -1, 1)

exit(gen.generate(PACKAGE, "pid", "Pid"))
```

7.5 ANNEXE E : Fonction réglage automatique des PID.

```
namespace autotune
{
double Ku = 0.;
double Tu = 0.;
double setpoint = 0.;
double state = 0.;
std::string ns = "/syropod/CL_tibia_joint/pid/";
int oscillation_count = 0;
int num_loops = 500; // Will look for oscillations for num_loops*loopRate
int initial_error = 0;
double p_ZN = 0.;
double i_ZN = 0.;
double d_ZN = 0.;
bool found_Ku = false;
std::vector<double> oscillation_times(10);
// Used to calculate Tu, the oscillation period
}
```

7.6 ANNEXE F : Architecture de YOLO V5

```
\label{Archi_yolo}
# parameters
nc: 22 # number of classes
depth_multiple: 1.0 # model depth multiple (to define)
width_multiple: 1.0 # layer channel multiple (to define)

# anchors ( To calculate using a python script)
anchors:
- [10,13, 16,30, 33,23] # P3/8 <= MODIFY VALUE
- [30,61, 62,45, 59,119] # P4/16 <= MODIFY VALUE
- [116,90, 156,198, 373,326] # P5/32 <= MODIFY VALUE

# YOLOv5 backbone
backbone:
# [from, number, module, args]
[[-1, 1, Focus, [64, 3]], # 0-P1/2
[-1, 1, Conv, [128, 3, 2]], # 1-P2/4
[-1, 3, C3, [128]],
[-1, 1, Conv, [256, 3, 2]], # 3-P3/8
[-1, 9, C3, [256]],
[-1, 1, Conv, [512, 3, 2]], # 5-P4/16
[-1, 9, C3, [512]],
[-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
[-1, 1, SPP, [1024, [5, 9, 13]]],
[-1, 3, C3, [1024, False]], # 9
]

# YOLOv5 head
head:
[[-1, 1, Conv, [512, 1, 1]],
[-1, 1, nn.Upsample, [None, 2, 'nearest']],
[[-1, 6], 1, Concat, [1]], # cat backbone P4
```

```
[-1, 3, C3, [512, False]], # 13

[-1, 1, Conv, [256, 1, 1]],
[-1, 1, nn.Upsample, [None, 2, 'nearest']],
[[-1, 4], 1, Concat, [1]], # cat backbone P3
[-1, 3, C3, [256, False]], # 17 (P3/8-small)

[-1, 1, Conv, [256, 3, 2]],
[[-1, 14], 1, Concat, [1]], # cat head P4
[-1, 3, C3, [512, False]], # 20 (P4/16-medium)

[-1, 1, Conv, [512, 3, 2]],
[[-1, 10], 1, Concat, [1]], # cat head P5
[-1, 3, C3, [1024, False]], # 23 (P5/32-large)

[[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
]
```

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] I. Jovancevic, J.-J. Orteu, T. Sentenac, et R. Gilblas, « Inspection d'un aéronef à partir d'un système multi-capteurs porté par un robot mobile », janv. 2015.
- [2] « Risques technologiques : la directive SEVESO et la loi Risques », Ministères Écologie Énergie Territoires. <https://www.ecologie.gouv.fr/risques-technologiques-directive-seveso-et-loi-risques> (consulté le 19 octobre 2022).
- [3] O. Prevention, « La prévention des risques des travaux en milieu confiné ». <https://www.officiel-prevention.com/dossier/protections-collectives-organisation-ergonomie/ventilation-aeration-filtrage-et-appareils-de-contrôle-d-atmosphère/la-prevention-des-risques-des-travaux-en-milieu-confine> (consulté le 19 octobre 2022).
- [4] « Prévention des risques industriels : un état des lieux », vie-publique.fr. <https://www.vie-publique.fr/eclairage/274283-la-prevention-des-risques-industriels-un-etat-des-lieux> (consulté le 19 octobre 2022).
- [5] « Carrières de Conflans-Sainte-Honorine Herblay - Boreally », Boreally Urban Exploration, 22 octobre 2006. <https://www.boreally.org/carriere/carrieres-conflans-st-honorine-herblay/> (consulté le 4 octobre 2022).
- [6] « NF EN ISO 17637 », Afnor EDITIONS. <https://m.boutique.afnor.org/fr-fr/norme/nf-en-iso-17637/contrôle-non-destructif-des-assemblages-soudés-contrôle-visuel-des-assembla/fa186772/58889> (consulté le 19 octobre 2022).
- [7] « ISO 9606-1/AC1 :2012 », Afnor EDITIONS. <https://m.boutique.afnor.org/fr-fr/norme/iso-9606-1-ac12012/rectificatif-technique-1-a-la-norme-iso-9606-1-de-juillet-2012/xs104909/121225> (consulté le 19 octobre 2022).
- [8] “Measurement of reinforcement corrosion in concrete adopting ultrasonic tests and artificial neural network,” *Construction and Building Materials*, vol. 177, pp. 125–133, Jul. 2018, doi : 10.1016/j.conbuildmat.2018.05.124.
- [9] « Inspection par Magnétoscopie (MPI) - Essais non destructifs », UberSpec. <https://uberspec.com/fr/ndt-solutions-contrôle-non-destructif/magnetoscopie-contrôle-non-destructif-cnd/> (consulté le 17 septembre 2022).

-
- [10] « Contrôle par thermographie – INSPECTION 3D NDT ». <https://ndtmaroc.com/2021/01/02/controle-par-thermographie/> (consulté le 17 septembre 2022).
- [11] « Qu'est-ce que l'inspection par particules magnétiques - Définition | Propriétés matérielles », Material Properties, 30 avril 2022. <https://material-properties.org/fr/quest-ce-que-linspection-par-particules-magnetiques-definition/> (consulté le 17 septembre 2022).
- [12] D. L. Atherton, "Remote field eddy current inspection," *IEEE Transactions on Magnetics*, vol. 31, no. 6, pp. 4142–4147, 1995.
- [13] « Contrôles volumiques radiographie », NDT Inspections. <https://ndt-inspections.fr/contrôles-volumiques-radiographie/> (consulté le 17 septembre 2022).
- [14] K. Bollas, D. Papasalouros, D. Kourousis, et A. Anastasopoulos, « ACOUSTIC EMISSION INSPECTION OF RAIL WHEELS. », *Journal of Acoustic Emission*, vol. 28, 2010.
- [15] D. Popescu, F. D. Anania, C. E. Cotet, et C. Amza, « Fully automated liquid penetrant inspection line simulation model for increasing productivity », *International Journal Simulation Modelling*, vol. 12, n 2, p. 82-93, 2013.
- [16] « Electrochemical Corrosion Testing | Electrochemical Corrosion Failure Analysis | Electrochemical Corrosion Material Analysis ». <https://www.mee-inc.com/hamm/electrochemical-corrosion-testing/> (consulté le 17 septembre 2022).
- [17] R. B. Santos, E. O. de Sousa, F. V. da Silva, S. L. da Cruz, et A. M. F. Fileti, « Detection and on-line prediction of leak magnitude in a gas pipeline using an acoustic method and neural network data processing », *Braz. J. Chem. Eng.*, vol. 31, p. 145-153, mars 2014, doi : 10.1590/S0104-66322014000100014.
- [18] M. Fahimipirehgalin, B. Vogel-Heuser, E. Trunzer, et M. Odenweller, « Visual Leakage Inspection in Chemical Process Plants Using Thermographic Videos and Motion Pattern Detection », *Sensors*, vol. 20, n 22, Art. n 22, janv. 2020, doi : 10.3390/s20226659.
- [19] D.-H. Jeong, C.-W. Jung, K.-S. Jung, et C.-K. Hong, « Dynamic characteristics of the PDLC-based electro-optic modulator for TFT LCD inspection », in *Optomechatronic Systems III*, oct. 2002, vol. 4902, p. 476-485. doi : 10.1117/12.467530.

-
- [20] J. J. Gertler, « Survey of model-based failure detection and isolation in complex plants », *IEEE Control Systems Magazine*, vol. 8, n 6, p. 3-11, déc. 1988, doi : 10.1109/37.9163.
- [21] R. S. Lim, H. M. La, Z. Shan, and W. Sheng, “Developing a crack inspection robot for bridge maintenance,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 6288–6293.
- [22] P. Demy, J.-C. Golinval, and D. Simon, “Détection d’endommagement de structures en matériaux composites CFRP par vibrothermographie,” p. 6, 2011.
- [23] Y. Zhang, X. Gao, and S. Katayama, “Weld appearance prediction with BP neural network improved by genetic algorithm during disk laser welding,” *Journal of Manufacturing Systems*, vol. 34, pp. 53–59, Jan. 2015, doi : 10.1016/j.jmsy.2014.10.005.
- [24] X. Hou and H. Liu, "Welding Image Edge Detection and Identification Research Based on Canny Operator," *2012 International Conference on Computer Science and Service System*, 2012, pp. 250-253, doi : 10.1109/CSSS.2012.70.
- [25] T. Merazi-Meksen, M. Boudraa, and B. Boudraa, “Mathematical morphology for TOFD image analysis and automatic crack detection,” *Ultrasonics*, vol. 54, no. 6, pp. 1642–1648, Aug. 2014, doi : 10.1016/j.ultras.2014.03.005.
- [26] P. Rea and E. Ottaviano, “Design and development of an Inspection Robotic System for indoor applications,” *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 143–151, Feb. 2018, doi : 10.1016/j.rcim.2017.06.005.
- [27] “Pilgrim Technology – When recognized inspectors meet certified robotic designer.” <https://www.pilgrim-technology.com/> (accessed Oct. 17, 2021).
- [28] A. Ortiz, F. Bonnin-Pascual, and E. Garcia-Fidalgo, *Vessel Inspection Assistance by means of a Micro-Aerial Vehicle : Control Architecture and Self-Localization Issues*. 2013.
- [29] « Côte d’Ivoire - Oryx Energies ». <http://www.oryxenergies.com/fr/notre-implantation/cote-ivoire> (consulté le 19 octobre 2022).
- [30] “Casablanca : effondrement d’une maison menaçant ruine suite aux fortes précipitations - La Vie éco.” <https://www.lavieeco.com/actualite-maroc/casablanca-effondrement-dune-maison-menacant-ruine-suite-aux-fortes-precipitations/> (accessed Oct. 17, 2021).

-
- [31] D. Griffin, “Guide To What Happens When A Drone Is Out Of Range.” <https://www.news6.com/story/5e9e0eeb7c6d3a5d636cab46/guide-to-what-happens-when-a-drone-is-out-of-range> (accessed Sep. 21, 2021)
- [32] Y.-J. Jo et M.-S. Jang, « Suggestion and Verification of Architecture for Collecting Fine Dust using Drone », *The Journal of the Korea institute of electronic communication sciences*, vol. 15, n 1, p. 125-132, 2020, doi : 10.13067/JKIECS.2020.15.1.125.
- [33] A. Seeni, B. Schäfer, and G. Hirzinger, “Robot Mobility Systems for Planetary Surface Exploration – State-of-the-Art and Future Outlook : A Literature Survey,” 2010, pp. 189–208. doi : 10.5772/6930.
- [34] C. Witting, M. Fehr, R. Bähnemann, H. Oleynikova, and R. Siegwart, “History-aware autonomous exploration in confined environments using mavs,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.
- [35] R. Li et al., “Opportunity rover localization and topographic mapping at the landing site of Meridiani Planum, Mars,” *Journal of Geophysical Research : Planets*, vol. 112, no. E2, 2007, doi : 10.1029/2006JE002776.
- [36] “Zephyr : A Landsailing Rover for Venus - NASA Technical Reports Server (NTRS).” <https://ntrs.nasa.gov/citations/20150000879> (accessed Nov. 12, 2021).
- [37] A. Schiele et al., « NanoKhod exploration rover-a rugged rover suited for small, low-cost, planetary lander mission », *IEEE robotics and automation magazine*, vol. 15, num 2, p. 96-107, 2008.
- [38] “JPL Robotics : Home Page.” <https://robotics.jpl.nasa.gov/> (accessed Sep. 28, 2021).
- [39] S. Ulamec et al., “A rover for the JAXA MMX Mission to Phobos,” in *70th International Astronautical Congress*, Washington D. C. USA, Oct. 2019, p. IAC-19. Accessed : Sep. 28, 2021. [Online]. Available : <https://elib.dlr.de/130664/>
- [40] N. G. Tsagarakis et al., “Walk-man : A high-performance humanoid platform for realistic environments,” *Journal of Field Robotics*, vol. 34, no. 7, pp. 1225–1259, 2017.
- [41] AQuRo : A Cat-like Adaptive Quadruped Robot With Novel Bio-Inspired Capabilities, Azhar Aulia Saputra (2021).
- [42] Y. Zhao, X. Chai, F. Gao, et C. Qi, « Obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III », *Robotics and Autonomous Systems*, vol. 103, p. 199-212, mai 2018, doi : 10.1016/j.robot.2018.01.007.

-
- [43] M. Bjelonic, N. Kottege, et P. Beckerle, « Proprioceptive control of an over-actuated hexapod robot in unstructured terrain », in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, oct. 2016, p. 2042-2049. doi : 10.1109/IROS.2016.7759321.
- [44] T. Ohnishi et T. Asakura, « Spider-Robot 8-Leg Cooperative Walking Velocity Control Strategy Based on Environmental Information », *JSME Int. J., Ser. C*, vol. 47, n 4, p. 1101-1107, 2004, doi : 10.1299/jsmec.47.1101.
- [45] « Collemboles et Diploures : description, cycle de vie, rôle écologique ». <https://www.bio-enligne.com/apterigotes/269-collemboles.html> (consulté le 17 septembre 2022).
- [46] M. Duman-Scheel et N. H. Patel, « Analysis of molecular marker expression reveals neuronal homology in distantly related arthropods », *Development*, vol. 126, n 11, p. 2327-2334, juin 1999, doi : 10.1242/dev.126.11.2327.
- [47] V. Dürr, J. Schmitz, et H. Cruse, « Behaviour-based modelling of hexapod locomotion : linking biology and technical application », *Arthropod Structure and Development*, vol. 33, n 3, p. 237-250, juill. 2004, doi : 10.1016/j.asd.2004.05.004.
- [48] C. Missbach et al., « Evolution of insect olfactory receptors », *eLife*, vol. 3, p. e02115, mars 2014, doi : 10.7554/eLife.02115.
- [49] G. M. Nelson, R. D. Quinn, R. J. Bachmann, W. C. Flannigan, R. E. Ritzmann, et J. T. Watson, « Design and simulation of a cockroach-like hexapod robot », in *Proceedings of International Conference on Robotics and Automation*, avr. 1997, vol. 2, p. 1106-1111 vol.2. doi : 10.1109/ROBOT.1997.614284.
- [50] H.-J. Weidemann, F. Pfeiffer, et J. Eltze, « A design concept for legged robots derived from the walking stick insect », in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, juill. 1993, vol. 1, p. 545-552 vol.1. doi : 10.1109/IROS.1993.583166.
- [51] J. S. Bay, « Design of the “army-ant” cooperative lifting robot », *IEEE Robotics Automation Magazine*, vol. 2, n 1, p. 36-43, mars 1995, doi : 10.1109/100.388293.
- [52] J. Albus, R. Bostelman, et N. Dagalakis, « The NIST SPIDER, A Robot Crane », *J Res Natl Inst Stand Technol*, vol. 97, n 3, p. 373-385, 1992, doi : 10.6028/jres.097.016.
- [53] X. Ding, Z. Wang, A. Rovetta, et J. M. Zhu, « Locomotion Analysis of Hexapod Robot », in *Climbing and Walking Robots*, B. Miripour, Éd. InTech, 2010. doi : 10.5772/8822.

-
- [54] P. Biswal et P. K. Mohanty, « Development of quadruped walking robots : A review », *Ain Shams Engineering Journal*, vol. 12, n 2, p. 2017-2031, juin 2021, doi : 10.1016/j.asej.2020.11.005.
- [55] B. L. Luk, A. A. Collie, D. S. Cooke, et S. Chen, « Walking and Climbing Service Robots for Safety Inspection of Nuclear Reactor Pressure Vessels », *Measurement and Control*, vol. 39, n 2, p. 43-47, mars 2006, doi : 10.1177/002029400603900201.
- [56] S. Hirose et E. F. Fukushima, « Development of mobile robots for rescue operations », *Advanced Robotics*, vol. 16, n 6, p. 509-512, janv. 2002, doi : 10.1163/156855302320535845.
- [57] « Obstacle avoidance for a hexapod robot in unknown environment | Springer-Link ». <https://link.springer.com/article/10.1007/s11431-016-9017-6> (consulté le 17 septembre 2022).
- [58] M.-G. Gaina, « Practical application using the hexapod robot for interior inspection of a Boeing 737-300 aircraft fuel tank », *INCAS Bulletin*, vol. 13, n 2, p. 59-66, 2021.
- [59] H. Montes, L. Mena, R. Fernández, J. Sarria, et M. Armada, « Inspection platform for applications in humanitarian demining », in *ASSISTIVE ROBOTICS : Proceedings of the 18th International Conference on CLAWAR 2015, 2016*, p. 446-453.
- [60] E. Karalarli, A. M. Erkmén, et I. Erkmén, « Intelligent gait synthesizer for hexapod walking rescue robots », in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, 2004*, vol. 3, p. 2177-2182.
- [61] F. Tedeschi et G. Carbone, « Design Issues for Hexapod Walking Robots », *Robotics*, vol. 3, p. 181-206, juin 2014, doi : 10.3390/robotics3020181.
- [62] P.-C. Lin, H. Komsuoğlu, et D. E. Koditschek, « Legged Odometry from Body Pose in a Hexapod Robot », in *Experimental Robotics IX*, vol. 21, M. H. Ang et O. Khatib, Éd. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006, p. 439-448. doi : 10.1007/1155224642.
- [63] U. Saranlı, M. Buehler, et D. E. Koditschek, « RHex : A Simple and Highly Mobile Hexapod Robot », *The International Journal of Robotics Research*, vol. 20, n 7, p. 616-631, juill. 2001, doi : 10.1177/02783640122067570.
- [64] R. J. Full, D. R. Stokes, A. N. Ahn, et R. K. Josephson, « Energy absorption during running by leg muscles in a cockroach », *Journal of Experimental Biology*, vol. 201, n 7, p. 997-1012, avr. 1998, doi : 10.1242/jeb.201.7.997.

-
- [65] J. T. Watson, R. E. Ritzmann, S. N. Zill, et A. J. Pollack, « Control of obstacle climbing in the cockroach, *Blaberus discoidalis*. I. Kinematics », *J Comp Physiol A*, vol. 188, n 1, p. 39-53, févr. 2002, doi : 10.1007/s00359-002-0277-y.
- [66] A. Roennau, G. Heppner, L. Pfozter, et R. Dillmann, « LAURON V : Optimized Leg Configuration for the Design of a Bio-Inspired Walking Robot », juill. 2013. doi : 10.1142/97898145255340071.
- [67] K. Kaltenbrunner et B. Renzl, « Dynamic Capabilities at the Civil Protection Exercise : Eu Taranis 2013 », *Research in Competence-Based Management*, vol. 7, p. 247-267, oct. 2014, doi : 10.1108/S1744-211720140000007009.
- [68] K. Hauser, T. Bretl, J.-C. Latombe, et B. Wilcox, « Motion Planning for a Six-Legged Lunar Robot », in *Algorithmic Foundation of Robotics VII : Selected Contributions of the Seventh International Workshop on the Algorithmic Foundations of Robotics*, S. Akella, N. M. Amato, W. H. Huang, et B. Mishra, Éd. Berlin, Heidelberg : Springer, 2008, p. 301-316. doi : 10.1007/978-3-540-68405-3_19.
- [69] S. M. LaValle, *Planning Algorithms*. Cambridge : Cambridge University Press, 2006. doi : 10.1017/CBO9780511546877.
- [70] R. Steindl et al., « Bruce – Design and Development of a Dynamic Hexapod Robot », arXiv :2011.00523 [cs], nov. 2020, Consulté le : 7 février 2022. [En ligne]. Disponible sur : <http://arxiv.org/abs/2011.00523>
- [71] F. Chen and M. R. Jahanshahi, "NB-CNN : Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion," in *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392-4400, May 2018, doi : 10.1109/TIE.2017.2764844.
- [72] W. Choi and Y. Cha, "SDDNet : Real-Time Crack Segmentation," in *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 8016-8025, Sept. 2020, doi : 10.1109/TIE.2019.2945265.
- [73] S. Milyaev et I. Laptev, « Towards reliable object detection in noisy images », *Pattern Recognit. Image Anal.*, vol. 27, n 4, p. 713-722, oct. 2017, doi : 10.1134/S1054661817040149.
- [74] J. Abanto-Bueno et J. Lambros, « Investigation of crack growth in functionally graded materials using digital image correlation », *Engineering Fracture Mechanics*, vol. 69, n 14-16, p. 1695-1711, 2002.

-
- [75] H. Hu, Q. Gu, et J. Zhou, « HTF : a novel feature for general crack detection », in 2010 IEEE International Conference on Image Processing, 2010, p. 1633-1636.
- [76] R. Cucchiara et F. Filicori, « The vector-gradient Hough transform », IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, n 7, p. 746-750, 1998.
- [77] X. Kong, Z. Zhang, L. Meng, et H. Tomiyama, « Machine Learning Based Features Matching for Fatigue Crack Detection », Procedia Computer Science, vol. 174, p. 101-105, janv. 2020, doi : 10.1016/j.procs.2020.06.063.
- [78] Y. LeCun, P. Haffner, L. Bottou, et Y. Bengio, « Object Recognition with Gradient-Based Learning », in Shape, Contour and Grouping in Computer Vision, D. A. Forsyth, J. L. Mundy, V. di Gesù, et R. Cipolla, Éd. Berlin, Heidelberg : Springer, 1999, p. 319-345. doi : 10.1007/3-540-46805-6_19.
- [79] A. Krizhevsky, I. Sutskever, et G. E. Hinton, « Imagenet classification with deep convolutional neural networks », Advances in neural information processing systems, vol. 25, 2012.
- [80] K. Yang, Y. Ding, P. Sun, H. Jiang, et Z. Wang, « Computer vision-based crack width identification using F-CNN model and pixel nonlinear calibration », Structure and Infrastructure Engineering, vol. 0, n 0, p. 1-12, oct. 2021, doi : 10.1080/15732479.2021.1994617.
- [81] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, et A. W. M. Smeulders, « Selective Search for Object Recognition », Int J Comput Vis, vol. 104, n 2, p. 154-171, sept. 2013, doi : 10.1007/s11263-013-0620-5.
- [82] H. O. Song, R. Girshick, S. Zickler, C. Geyer, P. Felzenszwalb, et T. Darrell, « Generalized Sparselet Models for Real-Time Multiclass Object Recognition », IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, n 5, p. 1001-1012, mai 2015, doi : 10.1109/TPAMI.2014.2353631.
- [83] J. Redmon, S. Divvala, R. Girshick, et A. Farhadi, « You Only Look Once : Unified, Real-Time Object Detection », arXiv :1506.02640 [cs], mai 2016, Consulté le : 10 mars 2022. [En ligne]. Disponible sur : <http://arxiv.org/abs/1506.02640>
- [84] M. Nie et C. Wang, « Pavement Crack Detection based on yolo v3 », in 2019 2nd International Conference on Safety Produce Informatization (IICSPI), 2019, p. 327-330.
- [85] W. Liu et al., « SSD : Single Shot MultiBox Detector », arXiv :1512.02325 [cs], vol. 9905, p. 21-37, 2016, doi : 10.1007/978-3-319-46448-0_2.

-
- [86] W. Wu, M. A. Qurishee, J. Owino, I. Fomunung, M. Onyango, et B. Atolagbe, « Coupling Deep Learning and UAV for Infrastructure Condition Assessment Automation », in 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, sept. 2018, p. 1-7. doi : 10.1109/ISC2.2018.8656971.
- [87] J. Redmon et A. Farhadi, « YOLOv3 : An Incremental Improvement », arXiv :1804.02767 [cs], avr. 2018, Consulté le : 10 mars 2022. [En ligne]. Disponible sur : <http://arxiv.org/abs/1804.02767>
- [88] J. Redmon et A. Farhadi, « YOLO9000 : Better, Faster, Stronger », arXiv :1612.08242 [cs], déc. 2016, Consulté le : 10 mars 2022. [En ligne]. Disponible sur : <http://arxiv.org/abs/1612.08242>
- [89] A. Bochkovskiy, C.-Y. Wang, et H.-Y. M. Liao, « YOLOv4 : Optimal Speed and Accuracy of Object Detection », arXiv :2004.10934 [cs, eess], avr. 2020, Consulté le : 10 mars 2022. [En ligne]. Disponible sur : <http://arxiv.org/abs/2004.10934>
- [90] Jocher G, Nishimura K, Mineeva T, Vilariño R. yolov5. Code repository <https://github.com/ultralytics/yolov5>. 2020 May.
- [91] C. Supeshala, « YOLO v4 or YOLO v5 or PP-YOLO? », Medium, 23 août 2020. <https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109> (consulté le 10 mars 2022).
- [92] Vakil-Baghmisheh, Mohammad-Taghi, Peimani, Mansour, Sadeghi, Morteza, Eteffagh, Mir Mohammad. (2007). Crack detection in beam-like structures using genetic algorithms. *Applied Soft Computing*. 8. 1150-1160. 10.1016/j.asoc.2007.10.003.
- [93] J. Park, Y.-C. Chen, Y.-J. Li, and K. Kitani, “Crack Detection and Refinement Via Deep Reinforcement Learning,” in 2021 IEEE International Conference on Image Processing (ICIP), Sep. 2021, pp. 529–533. doi : 10.1109/ICIP42928.2021.9506446.
- [94] R. Saveeth and S. Uma Maheswari, “HCCD : Haar-Based Cascade Classifier for Crack Detection on a Propeller Blade,” in First International Conference on Sustainable Technologies for Computational Intelligence, Singapore, 2020, pp. 421–432. doi : 10.1007/978-981-15-0029-9_33.
- [95] B. Wang, W. Zhao, P. Gao, Y. Zhang, and Z. Wang, “Crack Damage Detection Method via Multiple Visual Features and Efficient Multi-Task Learning Model,” *Sensors*, vol. 18, p. 1796, Jun. 2018, doi : 10.3390/s18061796.

-
- [96] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.
- [97] J. O'Rourke and K. Supowit, "Some np-hard polygon decomposition problems," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 181–190, 1983.
- [98] C. H. Papadimitriou, "The euclidean travelling salesman problem is np-complete," *Theoretical computer science*, vol. 4, no. 3, pp. 237–244, 1977.
- [99] [1]M. Petrlík, T. Báča, D. Hert, M. Vrba, T. Krajník, and M. Saska, "A Robust UAV System for Operations in a Constrained Environment," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, Feb. 2020, doi : 10.1109/LRA.2020.2970980.
- [100] C. Papachristos, S. Khattak and K. Alexis, "Autonomous exploration of visually-degraded environments using aerial robots," 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 775-780, doi : 10.1109/ICUAS.2017.7991510.
- [101] M. Giergiel, T. Buratowski, P. Małka, K. Kurc, P. Kohut, and K. Majkut, "The Project of Tank Inspection Robot," *KEM*, vol. 518, pp. 375–383, Jul. 2012, doi : 10.4028/www.scientific.net/KEM.518.375.
- [102] A. Lo Nigro, L. Brunori, G. Guassardo, and C. Panebianco, "Risk based approach to oil tankers and bulk carriers vetting," Mar. 2005.
- [103] A. Ortiz, F. Bonnin-Pascual, E. Garcia-Fidalgo, and J. P. Company-Corcoles, "Vision-Based Corrosion Detection Assisted by a Micro-Aerial Vehicle in a Vessel Inspection Application," *Sensors*, vol. 16, no. 12, Art. no. 12, Dec. 2016, doi : 10.3390/s16122118.
- [104] "Mines de fer de Lorraine." <http://tchorski.morkitu.org/3/lor-01.htm> (accessed Oct. 17, 2021).
- [105] R. Zimroz, M. Hutter, M. Mistry, P. Stefaniak, K. Walas, and J. Wodecki, "Why should inspection robots be used in deep underground mines?," in *Proceedings of the 27th International Symposium on Mine Planning and Equipment Selection-MPES 2018, 2019*, pp. 497–507.
- [106] Corke P., et al. : Mining robotics. In : Siciliano B., Khatib O. (eds) *Springer Handbook of Robotics*, Springer, Berlin, 1127–1150 (2008)
- [107] Green, J., Bosscha, P., Candy, L., Hlophe, K., Coetzee, S. and Brink, S. : Can a robot improve mine safety? Presented at *CAD/CAM, Robotics and Factories of the Future*, (2010)

-
- [108] Green, J. : Robots in mining. In : CSIR 3rd Biennial Conference : Science Real and Relevant, Pretoria, South Africa, 9 (2010)).
- [109] Bhondve, Trupti B. and Moresh Madhukar Mukhedkar. “Mobile Rescue Robot for Human Body Detection in Rescue Operation of Disaster.” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Energy* 3 (2014) : 9876-9882.
- [110] Séisme : la Turquie de nouveau durement touchée,” LCI. <https://www.lci.fr/international/video-seisme-la-turquie-de-nouveau-durement-touchee-2143728.html> (accessed Sep. 16, 2021)
- [111] « Tableau émissivités en thermographie - La Librairie Thermographique ». http://www.thethermographiclibrary.org/index.php?title=Tableau_en_thermographie (consulté le 15 avril 2022).
- [112] U. Nehmzow, “Quantitative analysis of robot–environment interaction—towards ‘scientific mobile robotics,’” *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 55–68, Jul. 2003, doi : 10.1016/S0921-8890(03)00010-1.
- [113] Erwin, A. Nevriyanto, and D. Purnamasari, “Image enhancement using the image sharpening, contrast enhancement, and Standard Median Filter (Noise Removal) with pixel-based and human visual system-based measurements,” in *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Aug. 2017, pp. 114–119. doi : 10.1109/ICECOS.2017.8167116.
- [114] L. Jin, W. Zhang, G. Ma, and E. Song, “Learning deep CNNs for impulse noise removal in images,” *Journal of Visual Communication and Image Representation*, vol. 62, pp. 193–205, Jul. 2019, doi : 10.1016/j.jvcir.2019.05.005.
- [115] A. Seeni, B. Schäfer, and G. Hirzinger, “Robot Mobility Systems for Planetary Surface Exploration – State-of-the-Art and Future Outlook : A Literature Survey,” 2010, pp. 189–208. doi : 10.5772/6930.
- [116] I. A. D. Nesnas et al., “Axel and DuAxel rovers for the sustainable exploration of extreme terrains,” *Journal of Field Robotics*, vol. 29, no. 4, pp. 663–685, 2012, doi : 10.1002/rob.21407.
- [117] G. Landis, « Robotic exploration of the surface and atmosphere of Venus », *Acta Astronautica*, vol. 59, p. 570-579, oct. 2006, doi : 10.1016/j.actaastro.2006.04.011.
- [118] « Cave Exploration Robot », Wevolver. <https://www.wevolver.com/specs/cave.exploration.robot> (consulté le 19 octobre 2022).

-
- [119] A. Schiele et al., « NanoKhod Exploration Rover - A Rugged Rover Suited for Small, Low-Cost, Planetary Lander Mission », *IEEE Robotics Automation Magazine*, vol. 15, n 2, p. 96-107, juin 2008, doi : 10.1109/MRA.2008.917888.
- [120] C. H. Lee, S. H. Kim, S. C. Kang, M. S. Kim, et Y. K. Kwak, « Double-track mobile robot for hazardous environment applications », *Advanced Robotics*, vol. 17, n 5, p. 447-459, janv. 2003, doi : 10.1163/15685530360663436.
- [121] J. Borenstein, M. Hansen, et A. Borrell, « The OmniTread OT-4 serpentine robot—design and performance », *J. Field Robotics*, vol. 24, n 7, p. 601-621, juill. 2007, doi : 10.1002/rob.20196.
- [122] M. Kovac, M. Fuchs, A. Guignard, J. Zufferey and D. Floreano, "A miniature 7g jumping robot," 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 373-378, doi : 10.1109/ROBOT.2008.4543236.
- [123] G. Ribak, « Insect-inspired jumping robots : challenges and solutions to jump stability », *Current Opinion in Insect Science*, vol. 42, p. 32-38, déc. 2020, doi : 10.1016/j.cois.2020.09.001.
- [124] J. Rogers, K. Page-Bailey, et R. Smith, « BounceBot : A One-Legged Jumping Robot », in *Towards Autonomous Robotic Systems*, vol. 10965, M. Giuliani, T. Assaf, et M. E. Giannaccini, Éd. Cham : Springer International Publishing, 2018, p. 52-63. doi : 10.1007/978-3-319-96728-8_5.
- [125] M. Bjelonic, P. K. Sankar, C. D. Bellicoso, H. Vallery, et M. Hutter, « Rolling in the Deep – Hybrid Locomotion for Wheeled-Legged Robots Using Online Trajectory Optimization », *IEEE Robot. Autom. Lett.*, vol. 5, n 2, p. 3626-3633, avr. 2020, doi : 10.1109/LRA.2020.2979661.
- [126] V. Klemm et al., "Ascento : A Two-Wheeled Jumping Robot," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 7515-7521, doi : 10.1109/ICRA.2019.8793792.
- [127] J. Monroy, J.-R. Ruiz-Sarmiento, F.-A. Moreno, F. Melendez-Fernandez, C. Galindo, and J. Gonzalez-Jimenez, "A Semantic-Based Gas Source Localization with a Mobile Robot Combining Vision and Chemical Sensing," *Sensors*, vol. 18, no. 12, Art. no. 12, Dec. 2018, doi : 10.3390/s18124174.
- [128] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment : theory,

-
- simulation and measurement,” *IEEE Computational Intelligence Magazine*, vol. 2, no. 2, pp. 37–51, May 2007, doi : 10.1109/MCI.2007.353419
- [129] J. Sawada, K. Kusumoto, Y. Maikawa, T. Munakata, and Y. Ishikawa, “A mobile robot for inspection of power transmission lines,” *IEEE Transactions on Power Delivery*, vol. 6, no. 1, pp. 309–315, Jan. 1991, doi : 10.1109/61.103753.
- [130] K. M. Wurm, C. Stachniss, et W. Burgard, « Coordinated multi-robot exploration using a segmentation of the environment », in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, sept. 2008, p. 1160-1165. doi : 10.1109/IROS.2008.4650734.
- [131] H. Hu, X. Ma, et Y. Shang, « A novel method for transformer fault diagnosis based on refined deep residual shrinkage network », *IET Electric Power Appl*, vol. 16, n 2, p. 206-223, févr. 2022, doi : 10.1049/elp2.12147.
- [132] « A Brief History of Industrial Robotics in the 20th Century ». <https://www.scirp.org/journal/paperinformation.aspx?paperid=90517> (consulté le 19 octobre 2022).
- [133] “Algorithme de Levenberg-Marquardt.” <https://www.statsoft.fr/concepts-statistiques/glossaire/l/levenberg.html> (accessed Oct. 19, 2021).
- [134] S. Bernardini et al., “A multi-robot platform for the autonomous operation and maintenance of offshore wind farms,” *Autonomous Agents and Multi-Agent Systems (AAMAS) 2020*, pp. 1696–1700, 2020.
- [135] M. Iida, D. Kang, M. Taniwaki, M. Tanaka, and M. Umeda, “Localization of CO2 source by a hexapod robot equipped with an anemoscope and a gas sensor,” *Computers and Electronics in Agriculture*, vol. 63, no. 1, pp. 73–80, Aug. 2008, doi : 10.1016/j.compag.2008.01.016.
- [136] F. Chaumette, S. Boukir, P. Bouthemy, D. Juvin. Structure from control- led motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5) :492-504, Mai 1996.
- [137] K. Nonami, R. K. Barai, A. Irawan, and M. R. Daud, “Kinematics, Navigation, and Path Planning of Hexapod Robot,” in *Hydraulically Actuated Hexapod Robots*, vol. 66, Tokyo : Springer Japan, 2014, pp. 85–104.
- [138] F. Chaumette, S. Hutchinson : Visual servo control, PartII : Advanced approaches.*IEEE Robotics and Automation Magazine*, 14(1) :109-118, March 2007.

-
- [139] F. Chaumette, “Visual Servoing,” in *Encyclopedia of Robotics*, M. H. Ang, O. Khatib, and B. Siciliano, Eds. Berlin, Heidelberg : Springer Berlin Heidelberg, 2020, pp. 1–9. doi : 10.1007/978-3-642-41610-1_104-1.
- [140] “The Image Jacobian | Lesson | Robot Academy,” May 08, 2017. <https://robotacademy.net.au/lesson/the-image-jacobian/> (accessed Nov. 03, 2021).
- [141] K. Qing-sheng, H. Ting, M. Zheng-da, and D. Xian-zhong, “Pseudo-inverse Estimation of Image Jacobian Matrix in Uncalibrated Visual Servoing,” in *2006 International Conference on Mechatronics and Automation*, Jun. 2006, pp. 1515–1520. doi : 10.1109/ICMA.2006.257399.
- [142] E. Knobloch, “Déterminants et élimination chez Leibniz / Determinants and elimination in Leibniz,” *Revue d’histoire des sciences*, vol. 54, no. 2, pp. 143–164, 2001, doi : 10.3406/rhs.2001.2116.
- [143] A. Dresden, “The fourteenth western meeting of the American Mathematical Society,” *Bull. Amer. Math. Soc.*, vol. 26, no. 9, pp. 385–397, Jun. 1920, doi : 10.1090/S0002-9904-1920-03322-7.
- [144] “Lees’ Loss Prevention in the Process Industries - 3rd Edition.” <https://www.elsevier.com/books/lees-loss-prevention-in-the-process-industries/lees/978-0-7506-7555-0> (accessed Nov. 04, 2021).
- [145] H. Greenhouse, R. Lowry, and B. Romenesko, “The Flow of Gases into Sealed Packages,” 2012, pp. 63–106. doi : 10.1016/B978-1-4377-7877-9.00004-0.
- [146] J. Burgués, V. Hernández, A. J. Lilienthal, and S. Marco, “Smelling Nano Aerial Vehicle for Gas Source Localization and Mapping,” *Sensors*, vol. 19, no. 3, Art. no. 3, Jan. 2019, doi : 10.3390/s19030478.
- [147] Azrad, Syaril, Kendoul, Farid , Nonami, Kenzo : Visual Servoing of Quadrotor Micro-Air Vehicle Using Color-Based Tracking Algorithm. *Journal of System Design and Dynamics*. 4. 255-268. 10.1299/jsdd.4.255, 2010.
- [148] J. Pages, C. Collewet, F. Chaumette and J. Salvi, ”An approach to visual servoing based on coded light,” *Proceedings 2006 IEEE International Conference on Robotics and Automation*, 2006. ICRA 2006., Orlando, FL, 2006, pp. 4118-4123.
- [149] Claudio, Giovanni, Spindler, Fabien, Chaumette, François : *Grasping by Romeo with visual servoing*, (2015).
- [150] Wurm, Kai M., Hornung A., Bennewitz M., Stachniss C. and Burgard W. (2010). “OctoMap : A Probabilistic, Flexible, and Compact 3D Map Representation for

Robotic Systems.” the University of Freiburg, Freiburg, Germany.

- [151] A.C. Sanderson, L.E. Weiss : Adaptive Visual Servo Control of Robots. In : Pugh A. (eds) Robot Vision. International Trends in Manufacturing Technology. Springer, Berlin, Heidelberg, 1983.
- [152] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. Kriegman, D. and Hager, G and Morse, A.S. The Confluence of Vision and Control, LNCIS Series, No 237, Springer-Verlag, pp.66-78, 1998.
- [153] P. I. Corke and S. A. Hutchinson, ”A new partitioned approach to image-based visual servo control,” in IEEE Transactions on Robotics and Automation, vol. 17, no. 4, pp. 507-515, Aug. 2001.
- [154] B.-G. Park, T.-H. Kim, and M.-J. Tahk, “Optimal impact angle control guidance law considering the seeker’s field-of-view limits,” Proceedings of the Institution of Mechanical Engineers, Part G : Journal of Aerospace Engineering, vol. 227, no. 8, pp. 1347–1364, Aug. 2013, doi : 10.1177/0954410012452367.
- [155] E. Malis, F. Chaumette and S. Boudet, ”2 1/2 D visual servoing,” in IEEE Transactions on Robotics and Automation, vol. 15, no. 2, pp. 238-250, April 1999.
- [156] Corke, P. I. , and Hutchinson, S. A. 1999. A new partitioned approach to image-based visual servo control . In Proceedings of the 31st International Symposium on Robotics and Automation.
- [157] Y. Huang and J. Su, “Visual Servoing of Nonholonomic Mobile Robots : A Review and a Novel Perspective,” IEEE Access, vol. 7, pp. 134968–134977, 2019, doi : 10.1109/ACCESS.2019.2941962.
- [158] A. H. A. Hafez, E. Cervera, and C. V. Jawahar, “Hybrid Visual Servoing by Boosting IBVS and PBVS,” in 2008 3rd International Conference on Information and Communication Technologies : From Theory to Applications, Apr. 2008, pp. 1–6, doi : 10.1109/ICTTA.2008.4530116.
- [159] K. Arshak, E. Moore, G. M. Lyons, J. Harris, and S. Clifford, “A review of gas sensors employed in electronic nose applications,” Sensor Review, vol. 24, no. 2, pp. 181–198, Jan. 2004, doi : 10.1108/02602280410525977.
- [160] D. W. Furry, “Methods for performing inspections and detecting chemical leaks using an infrared camera system,” US8193496B2, Jun. 05, 2012.

-
- [161] R. Fukasawa, "Terahertz Imaging : Widespread Industrial Application in Non-destructive Inspection and Chemical Analysis," *IEEE Transactions on Terahertz Science and Technology*, vol. 5, no. 6, pp. 1121–1127, Nov. 2015.
- [162] H. H. Hill and S. J. Martin, "Conventional analytical methods for chemical warfare agents," *Pure and Applied Chemistry*, vol. 74, no. 12, pp. 2281–2291, Jan. 2002, doi : 10.1351/pac200274122281.
- [163] Monroy J, Hernandez-Bennets V, Fan H, Lilienthal A, Gonzalez-Jimenez J. GADEN : A 3D Gas Dispersion Simulator for Mobile Robot Olfaction in Realistic Environments. *Sensors (Basel)*. 2017 Jun 23 ;17(7) :1479. doi : 10.3390/s17071479. PMID : 28644375 ; PMCID : PMC5539624.
- [164] J. Monroy, J.-R. Ruiz-Sarmiento, F.-A. Moreno, F. Melendez-Fernandez, C. Galindo, and J. Gonzalez-Jimenez, "A Semantic-Based Gas Source Localization with a Mobile Robot Combining Vision and Chemical Sensing," *Sensors*, vol. 18, no. 12, Art. no. 12, Dec. 2018, doi : 10.3390/s18124174.
- [165] Rasouli, F. ; Williams, T.A. Application of Dispersion Modeling to Indoor Gas Release Scenarios. *J. Air Waste Manag. Assoc.* 1995, 45, 191–195.
- [166] Holmes, N.S. ; Morawska, L. A review of dispersion modelling and its application to the dispersion of particles : an overview of different dispersion models available. *Atmos. Environ.* 2006, 40, 5902–5928.
- [167] Huang McQuaid, J. The box model of heavy gas dispersion : A useful and practical tool. *J. Occup. Accid.* 1984, 6, 253–261.
- [168] Leelőssy, Á. ; Molnár, F. ; Izsák, F. ; Havasi, Á. ; Lagzi, I. ; Mészáros, R. Dispersion modeling of air pollutants in the atmosphere : A review. *Cent. Eur. J. Geosci.* 2014, 6, 257–278.
- [169] Capelli, L. ; Sironi, S. ; Del Rosso, R. ; Guillot, J.M. Measuring odours in the environment vs. dispersion modelling : A review. *Atmos. Environ.* 2013, 79, 731–743.
- [170] Jasak, H., Jemcov, A., Tukovic, Z. (2007, September). OpenFOAM : A C++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics (Vol. 1000, pp. 1-20)*. IUC Dubrovnik Croatia.
- [171] Ansys Simulation Software Main Page. Available online : <http://www.ansys.com> (accessed on 22 June 2020).

-
- [172] Cabrita, G. ; Sousa, P. ; Marques, L. PlumeSim-Player/Stage Plume Simulator. In Proceedings of the ICRA 2010 Workshop for Networked and Mobile Robot Olfaction in Natural, Dynamic Environments, Örebro, Sweden, 3–8 May 2010.
- [173] B. Gerkey, R. T. Vaughan, et A. Howard, « The player/stage project : Tools for multi-robot and distributed sensor systems », in Proceedings of the 11th international conference on advanced robotics, 2003, vol. 1, p. 317-323.
- [174] J. Gonzalez-Jimenez, J. G. Monroy, et J. L. Blanco, « The multi-chamber electronic nose—An improved olfaction sensor for mobile robotics », *Sensors*, vol. 11, n 6, p. 6145-6164, 2011.
- [175] Eu, K.S. ; Yap, K.M. ; Tee, T.H. An Airflow Analysis Study of Quadrotor Based Flying Sniffer Robot. *Appl. Mech. Mater. Trans. Tech. Publ.* 2014, 627, 246–250.
- [176] J. Monroy, V. Hernandez-Bennetts, H. Fan, A. Lilienthal, and J. Gonzalez-Jimenez, “GADEN : A 3D Gas Dispersion Simulator for Mobile Robot Olfaction in Realistic Environments,” *Sensors*, vol. 17, no. 7, p. 1479, Jun. 2017, doi : 10.3390/s17071479.
- [177] M. D. Rocco, M. Reggente, and A. Saffiotti, “Gas source localization in indoor environments using multiple inexpensive robots and stigmergy,” in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sep. 2011, pp. 5007–5014, doi : 10.1109/IROS.2011.6094656.
- [178] H. Ishida, T. Ushiku, S. Toyama, H. Taniguchi, and T. Moriizumi, “Mobile robot path planning using vision and olfaction to search for a gas source,” in 2005 IEEE SENSORS, Oct. 2005, p. 4 pp.-, doi : 10.1109/ICSENS.2005.1597899.
- [179] G. Kowadlo and R. Russell, “Robot Odor Localization : A Taxonomy and Survey,” *I. J. Robotic Res.*, vol. 27, pp. 869–894, Aug. 2008, doi : 10.1177/0278364908095118.
- [180] E. Moraud and D. Martinez, “Effectiveness and Robustness of Robot Infotaxis for Searching in Dilute Conditions,” *Frontiers in neurorobotics*, vol. 4, p. 1, Mar. 2010, doi : 10.3389/fnbot.2010.00001.
- [181] S. Waphare, D. Gharpure, A. Shaligram, and B. Botre, “Implementation of 3-Nose Strategy in Odor Plume-Tracking Algorithm,” in 2010 International Conference on Signal Acquisition and Processing, Bangalore, India, Feb. 2010, pp. 337–341, doi : 10.1109/ICSAP.2010.81.
- [182] J. G. Monroy, A. J. Lilienthal, J.-L. Blanco, J. Gonzalez-Jimenez, and M. Trincavelli, “Probabilistic gas quantification with MOX sensors in Open Sampling Sys-

-
- tems—A Gaussian Process approach,” *Sensors and Actuators B : Chemical*, vol. 188, pp. 298–312, Nov. 2013, doi : 10.1016/j.snb.2013.06.053.
- [183] Khaliq, A. ; Pashami, S. ; Schaffernicht, E. ; Lilienthal, A. ; Hernandez Bennetts, V. Bringing Artificial Olfaction and Mobile Robotics Closer Together—An Integrated 3D Gas Dispersion Simulator in ROS. In *Proceedings of the 16th International Symposium on Olfaction and Electronic Nose (ISOEN)*, Dijon, France, 28 June–1 July 2015.
- [184] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario, “SPIRAL : A novel biologically-inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow,” *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 393–402, Apr. 2009, doi : 10.1016/j.robot.2008.07.004.
- [185] G. J. Garcia, J. A. Corrales, J. Pomares, et F. Torres, « Survey of Visual and Force/Tactile Control of Robots for Physical Interaction in Spain », *Sensors (Basel)*, vol. 9, numéro 12, p. 9689-9733, déc. 2009, doi : 10.3390/s91209689.
- [186] P. P. Neumann, V. H. Bennetts, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, “Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms,” *Advanced Robotics*, vol. 27, no. 9, pp. 725–738, Jun. 2013, doi : 10.1080/01691864.2013.779052.
- [187] Y. Kim and M. A. Minor, “Path Manifold-based Kinematic Control of Wheeled Mobile Robots Considering Physical Constraints,” *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 955–975, Sep. 2007, doi : 10.1177/0278364907081231.
- [188] A. Preumont, P. Alexandre, and D. Ghuys, “Gait analysis and implementation of a six leg walking machine,” in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, Jun. 1991, pp. 941–945 vol.2, doi : 10.1109/ICAR.1991.240551.
- [189] S. I. Greene and C. P. Spirito, “Interlimb Coordination During Slow Walking in the Cockroach : II. Effects of Cutting Thoracic Connectives,” *Journal of Experimental Biology*, vol. 78, no. 1, pp. 245–253, Feb. 1979.
- [190] Y. Wang, H. Lang and C. W. de Silva, ”A Hybrid Visual Servo Controller for Robust Grasping by Wheeled Mobile Robots,” in *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 757-769, Oct. 2010.
- [191] M. Taniwaki, M. Iida, D. Kang, M. Tanaka, T. Izumi, and M. Umeda, “Walking behaviour of a hexapod robot using a wind direction detector,” *Biosystems Engineering*, vol. 100, no. 4, pp. 516–523, Aug. 2008, doi : 10.1016/j.biosystemseng.2008.04.017.

-
- [192] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, “Speeded up detection of squared fiducial markers,” *Image and Vision Computing*, vol. 76, pp. 38–47, Aug. 2018, doi : 10.1016/j.imavis.2018.05.004.
- [193] Generation of fiducial marker dictionaries using mixed integer linear programming”, S. Garrido-Jurado, R. Muñoz Salinas, F.J. Madrid-Cuevas, R. Medina-Carnicer, *Pattern Recognition* :51, 481-491, 2016
- [194] G. Peng et al., “An Improved AMCL Algorithm Based on Laser Scanning Match in a Complex and Unstructured Environment,” *Complexity*, Dec. 11, 2018. <https://www.hindawi.com/journals/complexity/2018/2327637/> (accessed Feb. 26, 2021)
- [195] D. Fox, W. Burgard, F. Dellaert, et S. Thrun, « Monte Carlo Localization : Efficient Position Estimation for Mobile Robots », janv. 1999, p. 343-349.
- [196] C.-Y. Li, I.-H. Li, Y.-H. Chien, W.-Y. Wang, and C.-C. Hsu, “Improved Monte Carlo localization with robust orientation estimation based on cloud computing,” in 2016 IEEE Congress on Evolutionary Computation (CEC), Jul. 2016, pp. 4522–4527. doi : 10.1109/CEC.2016.7744365.
- [197] M. Aqel, M. H. Marhaban, M. I. Saripan, and N. Ismail, “Review of visual odometry : types, approaches, challenges, and applications,” *SpringerPlus*, vol. 5, Dec. 2016, doi : 10.1186/s40064-016-3573-7.
- [198] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte Carlo localization for mobile robots,” Feb. 1999, vol. 2, pp. 1322–1328 vol.2. doi : 10.1109/ROBOT.1999.772544.
- [199] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959, doi : 10.1007/BF01386390.
- [200] P. I. Corke, “High-performance visual servoing for robot end-point control,” in *Intelligent Robots and Computer Vision XII : Active Vision and 3D Methods*, Aug. 1993, vol. 2056, pp. 378–387. doi : 10.1117/12.150215.
- [201] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust Monte Carlo localization for mobile robots,” *Artificial Intelligence*, vol. 128, no. 1, pp. 99–141, May 2001, doi : 10.1016/S0004-3702(01)00069-8.
- [202] R. Gandhi, S. Nimbalkar, N. Yelamanchili, and S. Ponkshe, “Plant disease detection using CNNs and GANs as an augmentative approach,” in 2018 IEEE International

-
- Conference on Innovative Research and Development (ICIRD), May 2018, pp. 1–5.
doi : 10.1109/ICIRD.2018.8376321.
- [203] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” arXiv :1311.2524 [cs], Oct. 2014, Accessed : Nov. 05, 2021. [Online]. Available : <http://arxiv.org/abs/1311.2524>
- [204] G. Korotcenkov and B. Cho, “Chemical sensors selection and operation guide,” 2011, pp. 281–348.
- [205] T. Lochmatter and A. Martinoli, “Tracking Odor Plumes in a Laminar Wind Field with Bio-Inspired Algorithms,” *Spr. Tract. Adv. Rob.*, vol. 54, Nov. 2008, doi : 10.1007/978-3-642-00196-354.
- [206] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, et P. Dario, « SPIRAL : A novel biologically-inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow », *Robotics and Autonomous Systems*, vol. 57, n 4, p. 393-402, avr. 2009, doi : 10.1016/j.robot.2008.07.004.
- [207] A. T. Hayes, A. Martinoli, and R. M. Goodman, “Distributed odor source localization,” *IEEE Sensors Journal*, vol. 2, no. 3, pp. 260– 271, June 2002.
- [208] P. Ojeda, J. Monroy, et J. Gonzalez-Jimenez, « Information-Driven Gas Source Localization Exploiting Gas and Wind Local Measurements for Autonomous Mobile Robots », *IEEE Robot. Autom. Lett.*, vol. 6, n 2, p. 1320-1326, avr. 2021, doi : 10.1109/LRA.2021.3057290.
- [209] A. Valverde and P. Tsiotras, “Spacecraft Robot Kinematics Using Dual Quaternions,” *Robotics*, vol. 7, no. 4, Art. no. 4, Dec. 2018, doi : 10.3390/robotics7040064.
- [210] A. Censi, “An ICP variant using a point-to-line metric,” in 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, May 2008, pp. 19–25, doi : 10.1109/ROBOT.2008.4543181.
- [211] B. Tam, F. Talbot, R. Steindl, A. Elfes, and N. Kottege, “OpenSHC : A Versatile Multilegged Robot Controller,” *IEEE Access*, vol. 8, pp. 188908–188926, 2020, doi : 10.1109/ACCESS.2020.3031019.

-
- [212] Langlois, J., Mouchère H., Normand N. Viard-Gaudin C. (2016). “3D Orientation Estimation of Industrial Parts from 2D Images Using Neural Networks”. University of Nantes, Nantes, France.
- [213] Belter, D., Piotr S., (2011). “Integrated Motion Planning for a Hexapod Robot Walking on Rough Terrain”. University of Technology, Poznan, Poland.
- [214] Calisi D., Farinelli A., Iocchi L., Nardi D. (2007). “Multi-objective Exploration and Search for Autonomous Rescue Robots”. University of Roma “La Sapienza”, Roma, Italy.
- [215] Achtelika M., Bachrach A., Heb R., Prentice S. and Roy N. (2009). “Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-denied Indoor Environments”. Massachusetts Institute of Technology, Cambridge, USA.
- [216] Kruijff G.-J., Colas F., Svoboda T., Diggelen J., Patrick Balmer P., Pirri F., Worst R. (2012). “Designing Intelligent Robots for Human-Robot Teaming in Urban Search and Rescue”. Association for the Advancement of Artificial Intelligence.
- [217] Thilderkvist D., Svensson S. (2015). “Motion Control of Hexapod Robot Using Model-Based Design”. University of Lund, Sweden.
- [218] “urdf - ROS Wiki.” <http://wiki.ros.org/urdf> (accessed Oct. 03, 2021)
- [219] Murray D., Little J. (2000). “Using real-time stereo vision for mobile robot navigation”. Computer Science Department, University of British Columbia, Vancouver, Canada.
- [220] Angelova A., Matthies L., Helmick D., Perona P. (2007). “Fast Terrain Classification Using Variable-Length Representation for Autonomous Navigation”. California Institute of Technology, USA.

-
- [221] Biswas J, Veloso M. (2012). “Depth Camera Based Indoor Mobile Robot Localization and Navigation”. Carnegie Mellon University, Pennsylvania, USA.
- [222] Agrawal M., Konolige K. and Bolles R. (2007). “Localization and Mapping for Autonomous Navigation in Outdoor Terrains : A Stereo Vision Approach”. SRI International, USA.
Ghazouani H (2012). « Navigation visuelle de robots mobiles dans un environnement d’intérieur ». Université Montpellier II, Montpellier, France.
- [223] Baker C., Morris A., Ferguson D., Thayer S., Whittaker C., Omohundro Z., Reverte C., Whittaker W., Hahnel D., Thrun S. (2004). “A Campaign in Autonomous Mine Mapping”. Robotics Institute Carnegie Mellon University, Pittsburgh, USA.
- [224] Matabosch C., Salvi J., Fofi D. and Meriaudeau F. (2005). “Range image registration for industrial inspection”, University of Girona, Girona, Spain.
- [225] Schmidt A., Kasinski A.(2010). “The Visual SLAM System for a Hexapod Robot”. Institute of Control and Information Engineering, Poznan, Poland.
- [226] MacLeod, Charles N. , Summan, Rahul , Dobie, Gordon and Pierce, S. Gareth (2016) “Quantifying and improving laser range data when scanning industrial materials”. IEEE Sensors Journal. ISSN 1530-437X.
- [227] Prod’homme G., Richomme S., Reimeringer M., Kordek-Soenen M-A., Piquette B. , Macé Y. (2010), Rapport d’Etude N° DRA-09-102957-08289B, « Benchmark stockage en raffinerie , Maîtrise du vieillissement des installations industrielles », INERIS, France. 115 p
- [228] Alheib M. (2007). « Mise en sécurité des cavités souterraines d’origine anthropique : Surveillance – Traitement ». INERIS, France. 151 p
- [229] Krizhevsky A., Sutskever I., Hinton G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. University of Toronto, Toronto, Canada.

-
- Helmer S. , Lowe D. (2015). “Using Stereo for Object Recognition”. University of British Columbia, Vancouver, Canada.
- [230] Sathya Jose, S.L. and Dr. Sivaraman, K. (2017). “Recognition and extraction of noise from images using improved median filter”. *International Journal of Information Research and Review* Vol. 04, Issue, 02, pp.3657-3661,
- [231] Lubbe E. (2014). “State estimation of a hexapod robot using a proprioceptive sensory system”. North-West University, Potchefstroom, South Africa. 165 p.
- [232] Chinoy H.(2017). “Clothing Robots for rescue operations for radiation protection”. University of Saskatchewan, Saskatoon, Canada.
- [233] Aurelien Gonzalez (2013). Localisation par vision multi-spectrale. Application aux systèmes embarqués. Robotique [cs.RO]. INSA de Toulouse. Français.
- [234] Gongning Luo (2016). “A Deep Learning Network for Right Ventricle Segmentation in Short-Axis MRI”. University of Manchester, Manchester, UK.
- [235] Richard Socher (2012). “Convolutional-Recursive Deep Learning for 3D Object Classification”. Stanford University, Stanford, USA.
- [236] Yushi Chen (2014). “Deep Learning-Based Classification of Hyperspectral Data”. Harbin Institute of Technology, Harbin, China.
- [237] M. Konyev, F. Palis, Y. Zavgorodniy, A. Melnikov, A. Rudskiy, A. Telesh, U. Schmucker, and V.Rusin, “Walking Robot ‘ANTON’ : Design, Simulation, Experiments,” in *Advances In Mobile Robotics - Proceedings of the Eleventh International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2008, pp. 922–929.

-
- [238] L. Minati and A. Zorat, “A tree architecture with hierarchical data processing on a sensor-rich hexapod robot,” *Adv. Robot.*, vol. 16, no. 7, pp. 595–608, Jan. 2002.
- [239] U. Saranlı, A. Avci, and M. C. Ozturk, “A Modular Real-Time Fieldbus Architecture for Mobile Robotic Platforms,” *IEEE Trans. Instrum. Meas.*, vol. 60, no. 3, pp. 916–927, Mar. 2011.
- [240] Chen Zhou, Jiaolong yang, Chunshui Zhao, Gang Hua, “Fast, Accurate Thin-Structure Obstacle Detection for Autonomous Mobile Robots”, *IEEE*, 24 Aug. 2017.
- [241] A.C. Sanderson, Lee E. Weiss, ”Image-Based Visual Servo Control Of Robots,” *Proc. SPIE 0360, Robotics and Industrial Inspection*, 23 May 1983.
- [242] Wang Zijian, Wu Yimin, Yang Lichao, Thirunavukarasu Arjun, Evison Colin, Zhao Yifan. (2021). Fast Personal Protective Equipment Detection for Real Construction Sites Using Deep Learning Approaches. *Sensors*. 21. 3478. 10.3390/s21103478.
- [243] T. Salhi, « Régulateurs PID analogiques et digitaux », 2011.

RÉSUMÉ ÉTENDU

Le sujet de thèse porte sur la reconnaissance automatique de points d'intérêts pour un robot d'inspection dans un environnement contraint et dégradé. L'objectif de la thèse est d'effectuer des missions en autonomie en se basant sur les points d'intérêts visuels et chimiques détectés, une problématique bimodale. La combinaison des percepts visuels et chimiques permet d'optimiser la précision de localisation et assure une redondance d'informations. Le domaine d'étude concerne 3 cas d'application, le cas 1 : les zones de stockage d'hydrocarbure et les constructions navales, le cas 2 : les carrières souterraines et le cas 3 : les zones post-catastrophe. Dans cette étude des cas d'application, une méthode d'analyse contextuelle des cas est proposée et présentée afin d'analyser les contraintes des différents environnements complexes pour la solution robotique. La thèse regroupe donc différentes problématiques : l'étude des contraintes de l'environnement, le choix de la solution robotique, la navigation autonome et l'asservissement visuels et chimiques.

Suite à cette analyse contextuelle, un état de l'art est orienté sur la plateforme robotique terrestre pour déterminer la solution robotique la plus adaptée pour opérer dans les 3 cas d'application. Pour le sujet de thèse, les robots hexapodes ont été choisis pour leurs capacités à franchir les obstacles, leurs stabilités, et leurs capacités d'emport pour les capteurs, notamment. Une méthode est proposée pour atteindre la source du percept dans un environnement non structuré en s'appuyant sur les points d'intérêts visuels et chimiques. Dans cette étude, une approche d'asservissement visuel et chimique d'un robot hexapode est proposé.

Les systèmes d'asservissement basés sur l'image sont un schéma populaire dans la vision par ordinateur dans le domaine de la robotique mobile. La détection des objets est réalisée à l'aide d'un algorithme de réseau de neurones convolutifs de régions (YOLO v5) pour produire une inférence détectant les fissures, la corrosion, ainsi que plusieurs défauts industriels subtils liés aux métaux, difficilement perceptibles à l'œil nu. La base de données

est augmentée en reproduisant le bruit photométrique et géométrique pouvant être rencontré dans nos cas d'application (flou, contraste, etc.). Une matrice jacobienne visuelle est utilisée pour calculer l'orientation et la direction à prendre pour la partie commande de l'hexapode. L'idée de base derrière la matrice jacobienne visuelle est de modéliser la relation différentielle entre le système de caméra et le système de commande robotique pour détecter et suivre avec précision les points d'intérêt dans des environnements confinés. La navigation du robot est réalisée dans le repère monde. Les expérimentations menées sur un percept visuel simple, un QR code, a été réalisée et les résultats sont présentés et discutés.

La détection de la concentration du gaz est opérée à l'aide de 3 capteurs de gaz à oxydes métalliques (MOX) et d'un anémomètre pour la mesure du flux d'air. Un algorithme basé sur la concentration du gaz et la vitesse du vent est proposé, l'algorithme dit "grille", qui utilise la géométrie de la carte de l'environnement pour fournir une approche probabiliste par des estimations locales. L'efficacité du schéma proposé est d'abord démontrée par simulation. Enfin, un prototype d'hexapode est conçu, construit et développé en utilisant l'architecture logicielle ROS. L'hexapode présenté a réalisé une mission au sein d'un environnement industriel et à l'intérieur d'une construction navale, comprenant une série d'obstacles, des surbaux de 16 cm de hauteur. Les résultats de cette approche robotique sont présentés et commentés pour le cas 1, les environnements industriels et navals. La validation expérimentale de l'approche est présentée et discutée.

Titre : Reconnaissance automatique de points d'intérêts pour un robot d'inspection dans un environnement contraint et dégradé

Mot clés : Hexapode, inspection industrielle, asservissement visuel, asservissement chimique

Résumé : Le sujet de thèse porte sur la reconnaissance automatique de points d'intérêts (PI) pour un robot d'inspection dans un environnement contraint et dégradé. L'objectif de ces travaux de thèse est de développer une plateforme robotique capable d'effectuer des missions en autonomie en se basant sur des PI visuels et chimiques détectés, une problématique dite bimodale. La combinaison des percepts visuels et chimiques permet d'optimiser la précision de localisation et assure une redondance d'information. Le domaine d'étude concerne 3 cas d'application : le cas 1, l'inspection est réalisée dans un espace confiné (milieu industriel). Le cas 2, l'inspection est réalisée dans un environnement avec un risque avéré de perte de signal et à dominante rocheuse (mine, carrière souterraine). Le cas 3, l'inspection est réalisée dans un environnement ayant subi des déformations importantes et donc une géométrie des lieux d'inspection modifiée et chaotique (catastrophes naturelles de type séisme ou éboulement dans un environnement urbain). Dans cette étude, une méthode d'analyse contextuelle des cas est proposée et présentée afin d'analyser les contraintes des différents environnements complexes pour la solution robotique. La thèse regroupe donc différentes pro-

blématiques : l'étude des contraintes de l'environnement, le choix de la solution robotique, la navigation autonome et l'asservissement visuel et chimique. Suite à cette analyse contextuelle, un état de l'art est orienté sur la plateforme robotique terrestre pour déterminer la solution robotique la plus adaptée pour opérer dans les 3 cas d'application. Les robots hexapodes ont été choisis pour leurs capacités à franchir les obstacles, leurs stabilités, et leurs capacités d'emport pour les capteurs, notamment. Une méthode est proposée pour atteindre la source du percept dans un environnement non structuré en s'appuyant sur les PI visuels et chimiques. Pour l'évaluation de la méthodologie proposée, les PI visuels considérées sont de type QR code et la détection de la concentration d'un gaz concernant l'asservissement chimique. L'efficacité du schéma proposé est d'abord démontrée par des simulations. Enfin, un prototype d'hexapode est conçu, construit et développé en utilisant l'architecture logicielle ROS. L'hexapode développé a réalisé une mission au sein d'un environnement industriel et à l'intérieur d'une construction navale comprenant une série d'obstacles (cas 1 de l'étude). Les résultats de cette approche robotique sont enfin présentés, commentés et discutés.

Title: Automatic recognition of points of interest for an inspection robot in a constrained and degraded environment

Keywords: Hexapod, industrial inspection, visual servoing, chemical servoing

Abstract: The thesis subject concerns the automatic recognition of points of interest (PI) for an inspection robot in a constrained and degraded environment. The objective of this thesis work is to develop a robotic platform capable of carrying out autonomous missions based on detected visual and chemical PIs, a so-called bimodal problem. The combination of visual and chemical percepts optimizes localization accuracy and ensures information redundancy. The field of study concerns 3 application cases: case 1, the inspection is carried out in a confined space (industrial environment). Case 2, the inspection is carried out in an environment with a proven risk of loss of signal and predominantly rocky (mine, underground quarry). Case 3, the inspection is carried out in an environment that has undergone significant deformations and therefore a modified and chaotic geometry of the inspection sites (natural disasters such as earthquakes or landslides in an urban environment). In this study, a contextual case analysis method is proposed and presented in order to analyze the constraints of the different complex environments for the robotic solution. The thesis therefore brings together different issues: the study of environmental constraints, the choice of the robotic solution, autonomous navigation and visual and chemical servoing. Following this contextual analysis, a state of the art is oriented on the terrestrial robotic platform to determine the most suitable robotic solution to operate in the 3 application cases. The hexapod robots were chosen for their ability to overcome obstacles, their stability, and their carrying capacity for sensors, in particular. A method is proposed to reach the source of the percept in an unstructured environment by relying on visual and chemical PIs. For the evaluation of the proposed methodology, the visual PIs considered are of the QR code type and the detection of the concentration of a gas concerning chemical servoing. The effectiveness of the proposed scheme is first demonstrated by simulations. Finally, a hexapod prototype is designed, built and developed using the ROS software architecture. The developed hexapod carried out a mission within an industrial environment and inside a shipbuilding including a series of obstacles (case 1 of the study). The results of this robotic approach are finally presented, commented and discussed.