



**HAL**  
open science

# Contribution à l'autonomie des robots : vers des missions à garantie de performance incluant l'incertitude de localisation en environnement intérieur connu

Philippe Lambert

## ► To cite this version:

Philippe Lambert. Contribution à l'autonomie des robots : vers des missions à garantie de performance incluant l'incertitude de localisation en environnement intérieur connu. Robotique [cs.RO]. Université Montpellier, 2021. Français. NNT : 2021MONT018 . tel-03384111

**HAL Id: tel-03384111**

**<https://theses.hal.science/tel-03384111v1>**

Submitted on 18 Oct 2021

**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 POUR OBTENIR LE GRADE DE DOCTEUR  
DE L'UNIVERSITÉ DE MONTPELLIER**

**en Génie informatique, automatique et traitement du signal**

**École doctorale I2S**

**Unité de recherche LIRMM – UMR 5506**

**CONTRIBUTION À L'AUTONOMIE DES ROBOTS :  
VERS DES MISSIONS AUTONOMES  
À GARANTIE DE PERFORMANCE  
INCLUANT L'INCERTITUDE DE LOCALISATION  
EN ENVIRONNEMENT INTERIEUR CONNU**

**Présentée par Philippe LAMBERT  
Le 9 juin 2021**

**Sous la direction de Didier CRESTANI  
et de Lionel LAPIERRE**

**Devant le jury composé de**

**Didier CRESTANI, Professeur, Université de Montpellier**

**Jean-Philippe DIGUET, Directeur de Recherche, LAB-STICC**

**Lotfi JAIEM, Docteur, IADYS**

**Simon LACROIX, Directeur de Recherche, LAAS**

**Lionel LAPIERRE, Maître de Conférences HdR, Université de Montpellier**

**Jacques MALENFANT, Professeur, Sorbonne Université**

**Olivier PARODI, Docteur, Naval Group**

**René ZAPATA, Professeur, Université de Montpellier**

**Directeur de thèse**

**Rapporteur**

**Examineur**

**Rapporteur**

**Co-Directeur de thèse**

**Examineur**

**Examineur**

**Président**



**UNIVERSITÉ  
DE MONTPELLIER**

# REMERCIEMENT

Je tiens tout d'abord à remercier profondément Monsieur Didier CRESTANI mon directeur de thèse pour la confiance qu'il a placée en moi et pour son encadrement sans faille de la création du projet à sa réalisation présente. Merci aussi au co-encadrement de Monsieur Lionel LAPIERRE qui a permis un regard critique et toujours pertinent.

Je remercie également :

Messieurs Jean-Philippe DIGUET et Simon LACROIX d'avoir accepté la lourde tâche de relire cette thèse et d'en être rapporteurs. Les autres membres du jury Monsieur René ZAPATA, Monsieur Jacques MALENFANT, Monsieur Olivier PARODI et Monsieur Lotfi JAIEM pour avoir accepté d'évaluer ce travail.

Les membres de l'équipe EXPLORE qui m'ont accueilli avec bienveillance dont René ZAPATA, Karen GODARY, et plus particulièrement Sébastien DRUON pour ses connaissances et conseils sur les problématiques de vision.

Mes collègues doctorants passés et présents qui m'ont permis de détourner mon esprit sur d'autres questions et aussi d'être d'excellents « canards » Sylvain, Yohan, Tho, Adrien, Quentin, Rodolfo sans oublier Lotfi qui a posé les bases de ces travaux.

Le Service Handicap de l'Université de Montpellier que je ne saurais oublier et sans qui ce voyage n'aurait même pas commencé.

Les amis que je me suis fait depuis mon arrivée à Montpellier qui m'ont suivi tout le long de mon parcours et soutenu sans faillir quand j'en eus besoin, Sylvain, Clikoo, Nathan et Ludo futur confrère je l'espère. Diane sans qui ce manuscrit ne serait pas le même.

Je remercie ma famille qui a cru en moi bien avant et continuera bien après et m'a soutenu de toutes leurs possibilités.

Et enfin tous ceux qui ont participé d'une façon ou d'une autre à faire de moi la personne que je suis.

# Table des matières

<b>Table des figures</b>	<b>9</b>
<b>Liste des tableaux</b>	<b>13</b>
<b>1 INTRODUCTION GÉNÉRALE</b>	<b>17</b>
<b>2 AUTONOMIE ET GARANTIE DE PERFORMANCE</b>	<b>21</b>
2.1 Autonomie et performance . . . . .	21
2.2 De la performance à la garantie de performance : Positionnement général des travaux . . . . .	25
2.2.1 Le concept de performance . . . . .	25
2.2.2 Le pilotage de la performance . . . . .	26
2.2.3 Performance et Robotique . . . . .	26
2.2.3.1 Robotique industrielle . . . . .	27
2.2.3.2 Robotique mobile . . . . .	27
2.2.3.3 Mission robotique et performance . . . . .	28
2.2.4 Notre approche de la performance . . . . .	29
2.2.5 Garantie de performance - Cadre général de l'étude . . . . .	31
2.2.5.1 Garantie mono-axe (localisation) . . . . .	32
2.2.5.2 Garantie multi-axes . . . . .	34
2.2.5.3 Garantie par analyse formelle . . . . .	35
2.2.5.4 Notre approche de la garantie de performance . . . . .	36
2.3 La Méthodologie PANORAMA . . . . .	36
2.3.1 Quelques définitions . . . . .	36
2.3.2 P0 : Phase préparatoire . . . . .	38
2.3.2.1 Identification des inducteurs de performance . . . . .	39
2.3.2.2 Formulation des modèles prédictifs d'estimation de la performance	40
2.3.3 P1 : Phase préliminaire . . . . .	41
2.3.3.1 Représentation du scénario de la mission . . . . .	41
2.3.3.2 Les contraintes . . . . .	42
2.3.3.3 Projection des contraintes sur le scénario : Obtention du SMD .	43
2.3.4 P2 : Estimation hors ligne de la performance . . . . .	44
2.3.5 P3 : Gestion en ligne de la performance . . . . .	45
2.3.6 Apports et limitations . . . . .	46
2.4 Problématique scientifique de la thèse . . . . .	46
2.5 Conclusion . . . . .	47
<b>3 LA LOCALISATION EN ROBOTIQUE MOBILE : ÉLÉMENTS, PROBLÉMATIQUES ET APPROCHES</b>	<b>49</b>
3.1 Éléments de robotique mobile . . . . .	49
3.1.1 Quelques définitions . . . . .	49

3.1.2	Vue matérielle . . . . .	51
3.1.2.1	Les capteurs . . . . .	51
3.1.2.2	Actionneurs et effecteurs . . . . .	53
3.1.2.3	Organisations matérielle et spatiale . . . . .	54
3.1.3	Vue logicielle . . . . .	55
3.2	Problématiques de localisation . . . . .	57
3.3	Représentations de l'environnement . . . . .	59
3.3.1	Une typologie des environnements . . . . .	59
3.3.2	Les modèles d'environnement . . . . .	59
3.4	Les approches de localisation . . . . .	61
3.4.1	Localisation locale : Suivi de posture . . . . .	61
3.4.1.1	Méthodes non probabilistes . . . . .	61
3.4.1.2	Méthodes probabilistes . . . . .	63
3.4.2	Localisation globale . . . . .	65
3.4.2.1	Localisation par grille . . . . .	65
3.4.2.2	Filtrage particulière . . . . .	66
3.4.3	Autres approches . . . . .	67
3.4.3.1	Méthodes ensemblistes . . . . .	67
3.4.3.2	Méthodes basées sur les apparences . . . . .	69
3.4.4	Conclusion . . . . .	70
3.5	La localisation : Retour sur les incertitudes . . . . .	71
3.5.1	Mesure et incertitude . . . . .	71
3.5.2	Les incertitudes : De multiples origines . . . . .	72
3.5.3	Méthodes de localisation : incertitudes affichées . . . . .	74
3.5.3.1	Incertainces affichées et contexte expérimental . . . . .	75
3.5.4	Incertainces affichées : Enseignements . . . . .	77
3.5.5	Méthode, incertitude et garantie de Localisation . . . . .	79
3.6	Conclusion . . . . .	80
<b>4</b>	<b>CONTEXTE EXPÉRIMENTAL ET INCERTITUDES</b>	<b>81</b>
4.1	Localisation, pose et incertitudes : Vue d'ensemble . . . . .	81
4.1.1	Localisation et pose : Approche locale de suivi de posture . . . . .	82
4.1.2	Localisation et pose : Approche globale orientée grille . . . . .	84
4.1.3	Localisation, pose et incertitudes . . . . .	84
4.2	Contexte expérimental : Mission de patrouille . . . . .	85
4.2.1	Généralités . . . . .	86
4.3	Carte de l'environnement . . . . .	87
4.3.1	Vue d'ensemble . . . . .	87
4.3.2	Instrumentation de l'environnement et incertitudes . . . . .	88
4.4	La plateforme robotique : Vue matérielle . . . . .	90
4.4.1	Organisation Matérielle . . . . .	90
4.4.2	Modèle cinématique . . . . .	92
4.4.3	Organisation spatiale . . . . .	93
4.4.4	Capteurs . . . . .	95
4.4.4.1	Wattmètre embarqué . . . . .	95
4.4.4.2	Capteurs de choc . . . . .	96
4.4.4.3	Odomètre . . . . .	96
4.4.4.4	Ultrasons . . . . .	98
4.4.4.5	Laser . . . . .	100
4.4.4.6	Kinect 1 et marqueurs Aruco . . . . .	103

	A	Caractérisation statique . . . . .	104
	B	Caractérisation dynamique . . . . .	105
	C	Influence du champ de vision . . . . .	107
4.4.4.7	Kinect 2 et marqueurs Aruco . . . . .		110
	A	Caractérisation dynamique . . . . .	110
4.4.5	Actionneurs . . . . .		112
4.4.6	Conclusion . . . . .		113
4.5	La plateforme robotique : Vue logicielle . . . . .		113
4.5.1	Le middleware : CONTRACT . . . . .		114
4.5.2	Architecture logicielle : Modules, discrétisation et incertitude de localisation		115
	4.5.2.1	Les modules . . . . .	115
	4.5.2.2	Incertitude de Traitement Logiciel . . . . .	117
4.5.3	Conclusion . . . . .		119
4.6	Conclusion . . . . .		119

**5 PERFORMANCE DE LOCALISATION ET MODÈLES PRÉDICTIONNELS D'INCERTITUDE 121**

5.1	Performance de localisation : Démarche mise en œuvre . . . . .		121
5.2	ODO : Localisation proprioceptive odométrique . . . . .		124
	5.2.1	État de l'art . . . . .	124
	5.2.2	Phase 1 : Diagramme Causes-effets . . . . .	125
	5.2.3	Phase 2 : Erreurs systématiques . . . . .	126
	5.2.4	Phase 3-1 : Modèle d'erreur odométrique de position . . . . .	128
	5.2.5	Phase 3-3 : Incertitude de position : Évaluation . . . . .	130
	5.2.6	Phase 3-1 : Modèle d'erreur odométrique de rotation . . . . .	130
	5.2.7	Phases 3-3 et 4 : Incertitude de pose : Évaluation et Validation . . . . .	132
	5.2.8	Conclusion . . . . .	132
5.3	KIN-ARUCO : Localisation extéroceptive par amers artificiels verticaux . . . . .		133
	5.3.1	Phase 1 : Diagramme Causes-Effets . . . . .	133
	5.3.2	Phase 3-1 : Modèle d'erreur théorique . . . . .	134
	5.3.3	Phases 3-3 et 4 : Relevés expérimentaux et évaluation . . . . .	135
	5.3.4	Conclusion . . . . .	136
5.4	COMPAS : une méthode de détermination de l'orientation . . . . .		136
	5.4.1	Description de la méthode COMPAS . . . . .	136
	5.4.2	Méthode COMPAS : Modèle d'incertitude de localisation . . . . .	137
		5.4.2.1	Phase 1 : Diagramme Causes-Effets . . . . . 137
		5.4.2.2	Phase 3-3 : Construction expérimentale du modèle d'incertitude 138
	5.4.3	Conclusion . . . . .	139
5.5	COMPKIN . . . . .		139
	5.5.1	Phase 3-1 : Modèle d'erreur théorique . . . . .	139
	5.5.2	Phases 3-3 et 4 : Évaluation expérimentale et conclusion . . . . .	139
5.6	KINODO - COMPODOKIN . . . . .		141
	5.6.1	Phase 3-1 : Modèle d'erreur théorique . . . . .	141
	5.6.2	Phases 3-2 et 4 : Évaluation en simulation et conclusion . . . . .	142
5.7	KIN2ODO : Localisation extéroceptive par amers artificiels horizontaux . . . . .		143
5.8	LZA : Une nouvelle méthode de localisation globale . . . . .		144
	5.8.1	Description de la méthode LZA . . . . .	144
		5.8.1.1	Phases de mise en œuvre . . . . . 145
		A	Phase d'apprentissage (hors ligne) . . . . . 145
		B	Phase de transformation (hors ligne) . . . . . 146

	C	Phase de localisation en (x,y) (en ligne)	147
	D	Phase de localisation en $\theta$ (en ligne)	148
5.8.1.2		Algorithmes	148
5.8.1.3		Conclusion	149
5.8.2		Méthode LZA : Modèle d'incertitude de localisation	150
5.8.2.1		Phase 1 : Diagramme Causes-Effets	150
5.8.2.2		Phase 3-1 : Modèle d'erreur intervalliste	151
5.8.2.3		Phase 3-2 : Modèle d'incertitude : Évaluation statique en simulation	152
5.8.2.4		Phases 3-3 et 4 : Modèle d'erreur : Évaluation dynamique expérimentale	153
5.8.3		Méthode LZA-ODO : Amélioration de la relocalisation de LZA	155
5.8.3.1		Phase 3-2 : Évaluation en simulation	155
5.8.3.2		Phase 4 : Évaluation expérimentale du modèle d'erreur	157
5.8.4		Méthode LZA-ODO-COMPAS	158
5.8.4.1		Phase 4 : Évaluation expérimentale du modèle d'erreur	158
5.8.5		Conclusion	160
5.9		Conclusion	160

## **6 VERS UNE GARANTIE DE PERFORMANCE INCLUANT LA LOCALISATION : PRINCIPES DE MISE EN ŒUVRE 163**

6.1		Retour vers PANORAMA(SED)	163
6.2		La localisation : Un axe de performance particulier	164
6.3		Localisation et Algorithme de Viterbi	166
6.3.1		Formalisation du problème	166
6.3.2		Algorithme de Viterbi	170
6.3.3		Localisation, marge énergétique et algorithme de Viterbi	173
6.3.3.1		Garantie de Localisation	173
6.3.3.2		Maximisation de la marge énergétique	174
6.4		PANORAMA(SLED) : Garantie de performance SLED	176
6.4.1		Méthodologie finale SLED1	176
6.4.2		Méthodologie finale SLED2	178
6.4.3		Exemple didactique	179
6.5		Conclusion	182

## **7 VERS UNE GARANTIE DE PERFORMANCE INCLUANT LA LOCALISATION : SIMULATIONS ET EXPÉRIMENTATIONS 183**

7.1		Mission étudiée : Hypothèses, contraintes et objectifs	183
7.1.1		Mission de Patrouille : Description initiale	183
7.1.2		Tâches robotiques : schémas de contrôle, ressources matérielles et logicielles	184
7.1.3		Schémas de contrôle et ressources associées	184
7.1.4		Hypothèses de travail	184
7.1.5		Contraintes	185
7.1.5.1		Contraintes matérielles et expérimentales	186
7.1.5.2		Contraintes d'utilisation des méthodes de localisation	186
7.1.5.3		Contraintes/objectifs de performance	186
7.2		Présentation du simulateur	187
7.3		Utilisation des QR-Codes verticaux	188
7.3.1		PANORAMA(SED) : Scénario de Mission Détaillé	189
7.3.2		PANORAMA(SLED2 - QRC1) : ECHEC	193

7.3.3	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) . . . . .	193
7.3.3.1	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) - SMD : Axes Sécurité / Localisation . . . . .	193
7.3.3.2	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) - Viterbi : Détermination de la Solution d'Allocation de Ressources . . . . .	198
7.3.3.3	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Simulation . . . . .	199
7.3.3.4	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Analyse détaillée d'une mission . . . . .	201
7.3.3.5	Conclusion . . . . .	204
7.3.4	Conclusion . . . . .	204
7.4	Utilisation des QR-Codes verticaux et horizontaux . . . . .	204
7.4.1	PANORAMA(SED - QRC1 $\cup$ QRC3) : Scénario de Mission Détaillé . . . . .	205
7.4.2	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) . . . . .	206
7.4.2.1	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) - Viterbi : Détermination de la Solution d'Allocation de Ressources . . . . .	206
7.4.2.2	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Simulation . . . . .	208
7.4.2.3	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Analyse détaillée d'une mission . . . . .	209
7.4.2.4	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Analyse globale des missions réalisées . . . . .	212
7.4.3	Conclusion . . . . .	214
7.5	Conclusion . . . . .	214
<b>8</b>	<b>CONCLUSION GÉNÉRALE</b> . . . . .	<b>215</b>
8.1	Contexte des travaux . . . . .	215
8.2	Principaux résultats . . . . .	216
8.3	Limitations . . . . .	218
8.4	Perspectives . . . . .	220
<b>A</b>	<b>Énergie : Tables et abaqués des puissances consommées sur la batterie du robot</b> . . . . .	<b>225</b>
A.1	Puissance consommée : Modèle utilisé . . . . .	225
A.2	Ressources matérielles : Table des puissances consommées . . . . .	225
A.3	Puissance et énergie de déplacement . . . . .	226
A.3.1	Modèle de la puissance de déplacement . . . . .	226
A.3.2	Énergie de déplacement . . . . .	226
<b>B</b>	<b>Localisation : Tables de synthèse bibliographique</b> . . . . .	<b>227</b>
B.1	Méthodes de localisation locale . . . . .	227
B.2	Méthodes de localisation globale . . . . .	234
B.3	Méthodes de localisation par intervalle . . . . .	235
<b>C</b>	<b>Environnement : Référencement des QR-Codes ARUCO</b> . . . . .	<b>237</b>
C.1	Ensemble Initial : QRC1 . . . . .	237
C.2	QR-Codes verticaux ajoutés : QRC2 . . . . .	237
C.3	QR-Codes verticaux ajoutés : QRC3 . . . . .	237
	<b>Bibliographie</b> . . . . .	<b>241</b>





# Table des figures

2.1	Autonomie, décision et performance . . . . .	25
2.2	Pilotage et performance . . . . .	26
2.3	Indicateurs et objectifs de performance . . . . .	31
2.4	Tâches - Option d'implémentation de Tâche (OIT) et Alternative d'Implémen- tation (AI) (d'après [112]) . . . . .	38
2.5	La méthodologie PANORAMA . . . . .	38
2.6	Représentation d'une mission . . . . .	42
2.7	Classification des contraintes locales . . . . .	42
2.8	Projection des contraintes locales : Activités (d'après [112]) . . . . .	43
2.9	Principe d'allocation de ressources . . . . .	45
3.1	Le robot dans son environnement . . . . .	50
3.2	Les référentiels . . . . .	50
3.3	Exemples de capteurs : a) Proprioceptif - b) Intéroceptif - c) Extéroceptif . . . . .	52
3.4	Exemples de robots terrestres . . . . .	53
3.5	Exemples d'organisations matérielles : a) et b) Architecture électronique du robot martien <i>Pathfinder</i> - c) Plan d'un robot . . . . .	54
3.6	Principales classes architecturales robotiques . . . . .	56
3.7	Localisation et séquence Navigation-Guidage-Contrôle (inspiré de [168] [204]) . . . . .	57
3.8	Les problématiques de localisation (inspiré de [168]) . . . . .	58
3.9	Les modèles de représentation de l'environnement . . . . .	60
3.10	Suivi de posture . . . . .	62
3.11	Principe du filtre de Kalman . . . . .	64
3.12	Localisation par grille . . . . .	66
3.13	Localisation par filtrage particulière . . . . .	67
3.14	Algèbre ensembliste par intervalles . . . . .	68
3.15	Mesures : Illustration de quelques propriétés . . . . .	72
3.16	Localisation : Sources d'incertitudes . . . . .	74
4.1	Vue d'ensemble de l'environnement de la mission . . . . .	86
4.2	Carte enrichie de l'environnement de la mission . . . . .	88
4.3	Exemples de marqueurs visuels . . . . .	89
4.4	Référentiels Aruco . . . . .	90
4.5	Le robot patrouilleur . . . . .	91
4.6	Organisation matérielle du robot patrouilleur (d'après [112]) . . . . .	92
4.7	Structure cinématique d'un robot unicycle (d'après [25]) . . . . .	93
4.8	Plan de la plateforme robotique . . . . .	94
4.9	Positionnement des référentiels capteurs . . . . .	94
4.10	Le wattmètre embarqué Yocto-Watt . . . . .	96
4.11	Les détecteurs de choc . . . . .	96
4.12	Pioneer P3-Dx : Système de coordonnées internes - Référentiel Robot $\mathcal{R}_R$ . . . . .	97

4.13	Les banques de capteurs à ultrasons du Pioneer P3-DX . . . . .	98
4.14	Caractérisation des US : Exemples de relevés expérimentaux . . . . .	99
4.15	Pioneer P3-Dx : Erreur de mesure de distance des capteurs à ultrasons . . . . .	99
4.16	Carte de l'environnement : Zones où les capteurs à ultrasons sont indispensables . . . . .	100
4.17	Les capteurs laser du Pioneer P3-DX . . . . .	101
4.18	Courbes de réponse type des capteurs lasers du Pioneer P3-DX . . . . .	101
4.19	Carte de l'environnement : Zones impactant les mesures laser . . . . .	103
4.20	Le capteur Kinect v 1.0 à lumière structurée . . . . .	104
4.21	Caractérisation statique des marqueurs Aruco sur la Kinect v.1 . . . . .	104
4.22	Caractérisation statique des marqueurs Aruco sur la Kinect v.1 . . . . .	105
4.23	Caractérisation dynamique des marqueurs Aruco sur la Kinect v.1 . . . . .	106
4.24	Caractérisation dynamique des marqueurs Aruco sur la Kinect v.1 . . . . .	107
4.25	Décompte de la distance de recalage au cours d'une mission . . . . .	108
4.26	Détection d'un QR-Code à l'aide d'une Kinect . . . . .	109
4.27	Distance minimale possible et garantie d'un QR-Code excentré . . . . .	109
4.28	Le capteur Kinect v 2.0 à temps de vol . . . . .	110
4.29	Distance de détection de la Kinect 2 en dynamique . . . . .	111
4.30	Caractérisation dynamique de la Kinect 2 . . . . .	111
4.31	Histogramme de répartition des erreurs en Théta de la Kinect 2 . . . . .	112
4.32	Pioneer P3-Dx : Moteur Pittman 12V 9236 . . . . .	113
4.33	Rampe de commande du robot Pioneer . . . . .	113
4.34	CONTRACT : Schéma de principe de l'architecture logicielle . . . . .	114
4.35	Schémas d'exécution des boucles de Navigation-Guidage-Contrôle . . . . .	118
4.36	Erreur architecture et modèle de déplacement du robot . . . . .	119
5.1	Procédure de détermination d'un modèle d'erreur prédictif . . . . .	122
5.2	Diagramme causes-effets : Erreur de localisation odométrique . . . . .	126
5.3	Méthode UMBmark . . . . .	126
5.4	Évolution de l'erreur odométrique pour une trajectoire rectiligne . . . . .	130
5.5	Évolution de l'incertitude d'orientation odométrique pour une trajectoire rectiligne . . . . .	131
5.6	Comparaison entre l'erreur odométrique observée et celle du modèle pour une trajectoire rectiligne . . . . .	132
5.7	Diagramme causes-effets : Erreur de localisation Kin-Aruco . . . . .	133
5.8	Incertainitude de pose du modèle Kinect-Aruco : Comparaison des résultats expérimentaux avec le modèle théorique . . . . .	135
5.9	COMPAS : Principes méthodologiques . . . . .	137
5.10	Diagramme causes-effets : Erreur d'orientation Compas . . . . .	138
5.11	COMPAS : Histogramme des incertitudes observées . . . . .	138
5.12	Comparaison des méthodes KIN-ARUCO et COMPKIN . . . . .	140
5.13	Gain de COMPKIN versus KIN-ARUCO . . . . .	140
5.14	KINODO : Simulation de l'évolution de l'incertitude de pose . . . . .	142
5.15	COMPODOKIN : Simulation de l'évolution de l'incertitude de pose . . . . .	143
5.16	KIN2ODO : Simulation de l'évolution de l'incertitude de pose . . . . .	144
5.17	Environnement du robot - obstacles et mesures capteurs . . . . .	145
5.18	Méthode LZA : Exemples de pseudo-images . . . . .	146
5.19	Méthode LZA : Pseudo-images obtenues à l'issue de la phase de transformation . . . . .	146
5.20	Image binaire obtenue pour $\Xi_n = 0.05 \forall n$ . . . . .	147
5.21	Les profils $D_s$ et $D_r$ du robot simulé et réel . . . . .	148
5.22	LZA : Algorithmes . . . . .	149
5.23	Diagramme causes-effets : Erreur de localisation LZA . . . . .	151

5.24	LZA : Modèle d'erreur intervalliste . . . . .	152
5.25	LZA (localisation locale) : Évaluation statique et statistique . . . . .	152
5.26	LZA (localisation globale) : Évaluation statique et statistique . . . . .	153
5.27	LZA : Évaluation dynamique expérimentale . . . . .	154
5.28	LZA : Évaluation de la qualité de localisation . . . . .	155
5.29	LZA : Principe de l'algorithme de filtrage . . . . .	156
5.30	LZA-ODO : Relevés expérimentaux . . . . .	157
5.31	LZA-ODO-COMPAS : Relevés expérimentaux . . . . .	159
6.1	PANORAMA(SED) : Principales étapes . . . . .	164
6.2	Utilisation d'une méthode de localisation : Délais mis en jeu . . . . .	167
6.3	Impact du temps de préparation d'une méthode de localisation . . . . .	168
6.4	Scénario de mission et performance de Localisation . . . . .	169
6.5	Treillis et algorithme de Viterbi . . . . .	171
6.6	Algorithme de Viterbi : Scénario de Mission et délais de préparation . . . . .	175
6.7	PANORAMA(SLED1) : Principales étapes . . . . .	177
6.8	PANORAMA(SLED2) : Principales étapes . . . . .	178
6.9	PANORAMA(SED) : Situation initiale . . . . .	179
6.10	PANORAMA(SED) : Scénario SMD obtenu . . . . .	180
6.11	PANORAMA(SLED) : Situation initiale . . . . .	180
6.12	PANORAMA(SLED) : Filtrage garantissant la Localisation . . . . .	181
6.13	PANORAMA(SLED) : Scénario SMD obtenu . . . . .	181
7.1	Simulateur PANORAMA : Interface graphique . . . . .	188
7.2	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Évolution simulée de la performance de Pose . . . . .	200
7.3	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Évolution mesurée de la vitesse du robot . . . . .	201
7.4	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Évolution de la marge de temps . . . . .	202
7.5	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Évolution de la marge d'énergie . . . . .	203
7.6	PANORAMA(SLED2) (QRC1 $\cup$ QRC2) : Évolution mesurée de l'incertitude de Pose . . . . .	203
7.7	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Évolution simulée de la performance de Pose . . . . .	209
7.8	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Évolution mesurée de la vitesse du robot . . . . .	210
7.9	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Évolution de la marge de temps . . . . .	210
7.10	PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Évolution de la marge d'énergie . . . . .	211
7.11	PANORAMA(SLED2) (QRC1 $\cup$ QRC3) : Évolution mesurée de l'incertitude de pose . . . . .	211
7.12	PANORAMA(SLED2) (QRC1 $\cup$ QRC3) : Incertitude de pose mesurée et réalité terrain . . . . .	212
7.13	PANORAMA(SLED2) (QRC1 $\cup$ QRC3) : Bilan global des expérimentations . . . . .	213
7.14	Marges expérimentales pour les missions réussies . . . . .	213
A.1	Energie de déplacement du robot pour différentes distances . . . . .	226



# Liste des tableaux

2.1	Robotique : Quelques définitions de l'autonomie . . . . .	22
3.1	Localisation locale : Synthèse de l'analyse bibliographique . . . . .	75
3.2	Localisation globale : Synthèse de l'analyse bibliographique . . . . .	76
3.3	Méthodes de localisation par intervalle : Synthèse de l'analyse bibliographique . . . . .	77
4.1	Objectifs de la mission <i>Patrouiller</i> . . . . .	86
4.2	Paramètres du modèle cinématique d'un robot unicycle . . . . .	92
4.3	Paramètres de calibration constructeur du Pioneer P3-Dx . . . . .	98
4.4	Modules de liaison avec la partie matérielle . . . . .	115
4.5	Architecture : Modules de LOCALisation . . . . .	116
4.6	Architecture : Modules de DÉPLacement . . . . .	116
4.7	Architecture : Modules de ROTation . . . . .	116
4.8	Architecture : Module D'ANalyse d'Image . . . . .	117
4.9	PANORAMA : Modules de gestion de la mission . . . . .	117
5.1	LZA : Paramètres optimaux issus de l'algorithme génétique . . . . .	157
7.1	Objectifs de la mission de Patrouille . . . . .	184
7.2	Schémas de contrôle et ressources associées . . . . .	185
7.3	Hypothèses associées à la mission de Patrouille . . . . .	185
7.4	Contraintes matérielles et expérimentales de la mission de Patrouille . . . . .	186
7.5	Espaces de garantie de performance de la mission de Patrouille . . . . .	186
7.6	Objectifs de performance à garantir pour la mission de Patrouille . . . . .	187
7.7	PANORAMA SED : Scénario de Mission Détaillé et Solution d'Allocation de Ressources - Partie 1 - Aller . . . . .	190
7.8	PANORAMA SED : Scénario de Mission Détaillé et Solution d'Allocation de Ressources - Partie 2 - Retour . . . . .	191
7.9	PANORAMA(SED) : Utilisation des schémas de contrôle . . . . .	192
7.10	PANORAMA(SLED - QRC1 $\cup$ QRC2) : Scénario de Mission Détaillé - Partie 1 - Aller . . . . .	195
7.11	PANORAMA(SLED - QRC1 $\cup$ QRC2) : Scénario de Mission Détaillé - Partie 2 - Aller / Début retour . . . . .	196
7.12	PANORAMA(SLED - QRC1 $\cup$ QRC2) : Scénario de Mission Détaillé - Partie 3 - Fin retour . . . . .	197
7.13	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Solution d'Allocation de Ressources . . . . .	198
7.14	PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Analyse de la Solution d'Allocation de Ressources . . . . .	199
7.15	Performance énergétique à garantir avec QRC1 $\cup$ QRC3 . . . . .	205
7.16	PANORAMA(SED - QRC1 $\cup$ QRC3) : Scénario de Mission Détaillé et Solution d'Allocation de Ressources . . . . .	205

7.17	PANORAMA(SLED2 - QRC1 $\cup$ QRC3 ) : Scénario de Mission Détaillé et Solution d'Allocation de Ressources . . . . .	207
7.18	PANORAMA(SLED2 - QRC1 $\cup$ QRC3 ) : Analyse de la Solution d'Allocation de Ressources . . . . .	208
A.1	Puissance instantanée consommée . . . . .	225
B.1	Méthodes de localisation Locale sur un environnement de dimensions réduites . . . . .	229
B.2	Méthodes de localisation Locale sur un environnement de grandes dimensions . . . . .	230
B.3	Méthodes de localisation Locale en environnement 3D . . . . .	231
B.4	Méthodes de localisation Locale 1 . . . . .	232
B.5	Méthodes de localisation Locale 2 . . . . .	233
B.6	Méthodes de localisation Globale . . . . .	234
B.7	Méthodes de localisation par intervalle . . . . .	235
C.1	Table des poses des QR-Codes ARUCO de l'ensemble QRC1 . . . . .	238
C.2	Table des poses des QR-Codes ARUCO de l'ensemble QRC2 . . . . .	239
C.3	Table des localisation-3D des QR-Codes ARUCO de l'ensemble QRC3 . . . . .	240

# GLOSSAIRE

**A<sub>i</sub>** : Activité i

**AI** : Alternative d'implémentation

**COMPODOKIN** : Méthode de localisation utilisant l'odométrie entre la détection de deux marqueurs

**KINODO** : Méthode de localisation utilisant l'odométrie entre la détection de deux marqueurs Aruco v

**KIN2ODO** : Méthode de localisation utilisant l'odométrie entre la détection de deux marqueurs Aruco

**LZA** : Méthode de localisation globale basée sur les apparences et utilisant une grille de l'environnement

**LZA-ODO** : LZA utilisant l'odométrie entre deux estimations valides

**LZA-ODO-Compas** : LZA assistée d'un compas logiciel utilisant l'odométrie entre deux estimations v

**ODO** : Méthode de localisation basée sur l'odométrie

**OIT** : Option d'Implémentation de Tâche

**PANORAMA** : Performance and AutoNOmy using Resouce Allocation Management

**PANORAMA(SED)** : PANORAMA considérant les performances Sécuritaire, Énergie et Durée

**PANORAMA(SLED)** : PANORAMA considérant les performances Sécuritaire, Localisation, Énergie

**QRC1** : Ensemble initial de QR-Codes de la mission

**QRC2** : Ensemble de QR-Codes ajouté à la mission pour obtenir une solution sans KIN2ODO

**QRC3** : Ensemble de QR-Codes ajouté pour l'utilisation de KIN2ODO

**SAR** : Solution d'Allocation de Ressources

**SMD** : Scénario de Mission Détaillé

**X<sub>s</sub>** : Axe de l'abscisse curviligne de la trajectoire suivie par le robot

**Y<sub>s</sub>** : Axe normal à la trajectoire suivie par le robot



# NOTATIONS

$\mathcal{R}_R$  : Repère  $R$

$O_R$  : Origine du repère  $R$

$P|_R$  : Coordonnées de la pose d'un point  $P$  exprimées dans le repère  $R$

$O_{R(t)}|_M$  : Origine du repère  $R$  exprimé dans le repère  $M$  à l'instant  $t$

$MT_{\mathcal{R}_M|\mathcal{R}_{QR}}$  : Matrice de transformation homogène du repère  $QR$  au repère  $M$

$\theta_{\mathcal{R}_1|\mathcal{R}_2}$  : Angle du référentiel  $\mathcal{R}_2$  par rapport au référentiel  $\mathcal{R}_1$

# Chapitre 1

## INTRODUCTION GÉNÉRALE

La notion d'autonomie est un concept loin d'être facile à cerner et à caractériser. De nombreux courants disciplinaires tant philosophiques que scientifiques ont cherché à la circonscrire. La capacité décisionnelle dont se doivent d'être dotés les systèmes autonomes en est sans nul doute le marqueur central. L'autonomie se déploie au sein d'un environnement avec pour objectif la réalisation d'actions jugées contextuellement, localement et globalement pertinentes dans le temps et dans l'espace, par le système. La réalisation d'une action implique une dimension physique (*manière d'agir sur quelque chose ou quelqu'un*<sup>1</sup>) qui suppose que le système est capable de se positionner au sein de l'environnement dans lequel il évolue. La localisation joue donc un rôle central pour le déploiement de l'autonomie.

La capacité de localisation paraît naturelle aux êtres humains. Nos cinq sens nous renvoient des informations allothétiques (extéroceptives) sur notre environnement que notre cerveau traite en permanence pour nous permettre de nous mouvoir et d'interagir avec les objets ou personnes qui nous entourent. Même si la vision reste la modalité dominante, il ne faut pas minorer l'importance du rôle des informations idiothétiques (intéroceptives) avec la proprioception qui permet de localiser la position et l'orientation des différentes parties de notre corps dans l'espace, et le système vestibulaire qui nous renseigne sur nos mouvements (accélération) [40].

Si l'on y regarde de plus près, un individu construit une carte cognitive de son environnement qui correspond à la représentation interne qu'il se fait de son organisation, et qui évolue en fonction de notre expérience. Ces cartes mentales nous permettent à tout instant (dynamiquement) de connaître notre localisation et de planifier nos actions [92].

Notre représentation de l'espace fait appel à plusieurs référentiels en fonction du point de vue retenu [142] : Allocentrique (centré sur l'environnement), et égocentrique (centré sur notre corps). Ce dernier se décompose lui même en fonction de la perception qu'a l'individu des mouvements à mettre en œuvre [96]. On distingue l'espace de saisie (préhension directe), du champ proche (utilisation d'un outil) et de l'espace lointain (locomotion nécessaire).

Nous construisons ou nous disposons donc d'une représentation mentale des lieux dans lesquels nous évoluons et en fonction du contexte de nos actions nous recherchons une localisation plus ou moins précise. Nous sommes aussi capables de nous déplacer à l'estime dans le noir sur de courtes distances dans des lieux que nous connaissons, seulement à l'aide de notre perception proprioceptive, grâce à notre mémoire locomotrice. Cependant, même lorsque notre vision peut être employée, en l'absence d'éléments discriminants auxquels nous raccrocher, nous sommes perdus, incapables de nous localiser. En effet pour planifier nos actions nous avons besoin de nous appuyer sur une organisation hiérarchisée de connaissances spatiales de type repères, routes (liaisons entre repères) et survol (carte mentale de l'environnement vue de dessus) [216].

---

1. Dictionnaire de l'académie française

Il existe évidemment de nombreuses analogies entre les entités biologiques que nous sommes et les systèmes autonomes tels que les robots mobiles qui nous intéresseront dans le cadre de ce manuscrit. Pour eux aussi la capacité de localisation est un marqueur incontournable de l'autonomie. C'est un domaine de la robotique des plus prolifiques pour lequel de très nombreuses approches permettant au robot de se repérer dans son environnement ont été proposées. Les robots sont dotés de capteurs lui permettant d'appréhender, tout comme nos sens, l'environnement dans lequel il évolue, ainsi que son propre mouvement. Cet aspect matériel est l'une des rares dimensions du problème de localisation pour lequel les systèmes artificiels peuvent se révéler très supérieurs aux êtres humains. En effet nos organes sensoriels possèdent des limites de fonctionnement et de sensibilité qui peuvent être largement dépassées par les capteurs. C'est ensuite que la machine va se révéler moins performante. Il va falloir fusionner la grande quantité d'informations recueillies, plus ou moins bruitées, et être capable d'interpréter ces données à sémantique faible pour conclure quand à la localisation du robot au sein de son environnement. La mémorisation et/ou la construction de la représentation de ce dernier, même si plusieurs niveaux de granularité ont été proposés, demeure encore très élémentaire par rapport aux informations cognitives très variées que les êtres humains sont capables d'intégrer pour se localiser. Enfin le traitement de cette masse de données peut aussi induire un coût de calcul important parfois incompatible avec une utilisation réellement opérationnelle et continue d'une technique de localisation. Malgré tout, on peut sans doute considérer qu'actuellement on dispose en robotique d'un très large panel de méthodes de localisation aux performances et limitations assez largement éprouvées.

La performance d'une méthode de localisation est bien évidemment directement reliée à la précision avec laquelle le robot est capable de se localiser. Celle-ci dépend clairement de la fiabilité des informations capteurs recueillies, et donc souvent du coût de ces derniers, mais aussi de leur adéquation à l'environnement. Le choix de la méthode de localisation retenue et des principes théoriques sous-jacents influent aussi significativement sur la qualité de la localisation. La nature intrinsèque d'une méthode la rend plus ou moins performante en fonction des capteurs mobilisés, des ressources de calcul nécessaires et disponibles, des limites de principe associées (grille), et de l'environnement traversé. Il existe donc une adéquation méthode/environnement. Dans les articles scientifiques, cette précision est le plus souvent évaluée de façon factuelle, a posteriori, dans un environnement donné. Cependant, comme nous le verrons dans la suite de ce document, les conditions d'obtention de la réalité terrain ne sont pas toujours clairement explicitées, et il faut bien distinguer si l'on s'intéresse à un positionnement relatif par rapport à un objet, ou, plus généralement, à un repérage global au sein d'une carte. Quoiqu'il en soit, en raison du lien précision/action, il est indispensable de pouvoir garantir que le système robotique est réellement localisé en deçà d'une marge d'erreur acceptable pour pouvoir engager l'action envisagée. Certes, il existe des méthodes dites ensemblistes ou intervallistes permettant de vérifier si les contraintes de localisation visées le sont effectivement. Mais, à notre connaissance, ces travaux ne procurent, là encore, dans leur très large majorité, qu'une évaluation à posteriori de l'espace au sein duquel se trouve avec certitude le robot, sans pour autant pouvoir assurer, à priori, que ces dimensions vérifient les contraintes de localisation imposées.

Mais évidemment une mission robotique autonome ne peut se restreindre à la seule dimension de la Localisation. On doit entre-autres être capable de prendre en compte l'aspect sécuritaire et énergétique, sans oublier des points de vue plus utilisateur tels que la durée de la mission. Toutes ces dimensions sont corrélées et le choix de certaines méthodes de localisation impactera nécessairement les autres dimensions de la performance notamment l'énergie. L'équipe Explore au sein de laquelle ont été déployés les travaux présentés dans ce manuscrit a développé une méthodologie dénommée PANORAMA qui cherche à planifier l'utilisation et

le paramétrage des ressources matérielles et logicielles d'un robot mobile autonome de façon à garantir la satisfaction des contraintes de performance imposées, tout au long d'une mission s'exécutant dans un environnement connu. Elle s'appuie sur la mise en place de modèles de performance prédictifs et sur le suivi en ligne de l'évolution des marges de performance estimées. Cependant les premiers travaux présentés dans [112], bien que prenant en compte les dimensions sécuritaire, énergétique et durée d'une mission, n'appréhendaient la dimension localisation que de façon arbitraire et heuristique en postulant que les méthodes sélectionnables contextuellement permettaient effectivement de satisfaire les contraintes de localisation imposées.

Le travail présenté dans ce manuscrit est centré sur la garantie de localisation qui doit être assurée lors d'une mission robotique. Pour ce faire nous allons donc chercher à établir des modèles de localisation prédictifs nous permettant d'assurer la garantie recherchée tout au long d'une mission robotique autonome. Comme nous le montrerons, en raison de la spécificité de l'axe de performance localisation, les choix effectués auront un impact sur la dimension énergétique d'une mission, ce qui nous conduira à modifier ou plutôt à enrichir la démarche méthodologique initialement proposée dans PANORAMA.

Ce document est décomposé en 6 parties. Les 3 dernières détaillent les contributions majeures de cette thèse.

Le **second chapitre** s'intéresse plus particulièrement aux concepts d'autonomie et de performance qui soutendent globalement l'ensemble des travaux présentés. L'autonomie, qui n'est pas des plus facile à définir, est une notion multiforme étudiée dans de nombreux champs disciplinaires. La performance quand-à-elle bien que couramment employée est rarement définie clairement. C'est aussi une notion polymorphe que nous chercherons à caractériser de façon générale puis plus particulièrement dans le champ scientifique de la robotique. Nous nous focaliserons alors plus avant sur la vision adoptée au sein de l'équipe Explore et sur son lien avec la notion de garantie, tant dans sa dimension mono que multi points-de-vue. Les principales étapes de la méthodologie PANORAMA proposée dans [112] et qui constitue à la fois le cadre et le point de départ de ce travail, seront alors résumées. Pour terminer les problématiques scientifiques adressées dans le cadre de ces travaux seront précisées.

Le **troisième chapitre** effectue un état de l'art rapide autour de la robotique et des méthodes de localisation. Dans un premier temps, après avoir rappelé et défini les notions de repères et de matrices de transformation, un système robotisé est présenté au travers de ses vues matérielle et logicielle. Dans un second temps, les problématiques centrales de localisation sont explicitées puis les principales approches de localisation sont résumées. Enfin, la notion d'incertitude et son lien avec la localisation sont abordés.

Le **quatrième chapitre** revisite le chapitre précédent à l'aune de la plateforme robotique utilisée et de la mission robotique autonome visée dans le cadre de ce manuscrit, pour identifier et caractériser les incertitudes impactant le processus de localisation. Après avoir défini les différents référentiels qui devront être considérés et manipulés, nous établirons les équations de changement de référentiels mises en jeu dans une logique de suivi de posture ou de localisation absolue pour identifier la pose d'un robot. Celles ci seront alors analysées au regard des incertitudes qu'elles véhiculent. Après avoir présenté la mission de patrouille en environnement intérieur qui sous-tendra l'évaluation de nos travaux, les incertitudes liées à notre connaissance de l'environnement et aux dimensions tant matérielles que logicielles sont caractérisées en faisant appel à une vue ensembliste nécessaire pour adresser la dimension de garantie imposée par

la problématique adressée.

Le **cinquième chapitre** est consacré à la construction et à la validation des différents modèles prédictifs de Localisation associés à l'utilisation d'une ou plusieurs méthodes. En s'appuyant à la fois sur les incertitudes établies dans le chapitre précédent, et sur un ensemble de relevés expérimentaux, des modèles prédictifs d'évolution de l'incertitude de pose du robot sont construits pour un ensemble de méthodes. A celles classiques relevant de l'odométrie ou de recalage global par vision de QR-Code, s'ajoute aussi la mise en œuvre d'un compas logiciel utilisant les données laser. Par ailleurs une nouvelle méthode de localisation absolue LZA, faisant appel à une grille est présentée, évaluée, enrichie et caractérisée pour pouvoir être utilisée dans le cadre de PANORAMA.

Le **sixième chapitre** est une des parties centrales de ce document. Il présente dans un premier temps les spécificités de l'axe Localisation par rapport aux autres axes de performance. Dans un second temps il détaille l'algorithme de Viterbi qui a été retenu pour supporter le déploiement de l'approche de garantie de Localisation et qui permet de s'inscrire dans une logique, jusque là inenvisageable, de maximisation de la marge énergétique. Son intégration dans la méthodologie PANORAMA est présentée et son comportement est explicité sur un exemple didactique.

Le **septième chapitre** présente et commente la mise en œuvre de la méthodologie PANORAMA avec prise en compte de l'axe Localisation sur l'exemple de Mission de Patrouille qui nous sert de fil rouge. Dans un premier temps, après avoir présenté le simulateur développé, les plans de mission obtenus après la mise en œuvre simulée d'une approche PANORAMA avec et sans prise en compte explicite de la localisation sont détaillés, explicités puis comparés. Dans un second temps les résultats expérimentaux sont présentés et commentés.

Pour terminer la **conclusion générale** rappelle tout d'abord les principaux résultats présentés dans ce document avant de chercher à en tirer un certain nombre de réflexions et d'enseignements. Nous reviendrons sur la réelle capacité de localisation des robots mobiles et sur son coût d'évaluation et de mise en œuvre. Nous nous projeterons sur l'intérêt de l'approche PANORAMA dans le cadre d'explorations robotiques autonomes. Nous nous interrogerons enfin sur la signification que peut recouvrir la garantie de localisation dans certains contextes expérimentaux et sur les moyens de la mettre en œuvre.

# Chapitre 2

## AUTONOMIE ET GARANTIE DE PERFORMANCE

Ce chapitre permet de préciser le contexte scientifique général de cette étude. La première partie rappelle tout d’abord le cheminement qui mène de l’autonomie à la performance. Nous précisons alors ce que désigne, pour nous, une mission à garantie de performance. Les travaux menés au sein de l’équipe EXPLORE sur l’autonomie comportementale sont ensuite résumés. Ils visent à permettre le déploiement de missions robotiques autonomes, tolérantes aux fautes, à garantie de performance. Bien qu’offrant certaines garanties à l’utilisateur, l’approche proposée souffre d’un ensemble de limitations qui sont alors présentées. Parmi ces dernières nous nous focaliserons dans la suite de ce document sur la garantie.

### 2.1 Autonomie et performance

La notion d’autonomie est apparue dans l’antiquité pour désigner des citoyens qui édictaient leurs propres (*autos*) lois (*namos*). Depuis le champ lexical de ce terme s’est étendu pour être utilisé dans des disciplines très variées telles que la philosophie, la psychologie ou la robotique et donc pour être appliqué à des systèmes tant biologiques qu’artificiels.

L’approche philosophique Kantienne [124] considère que l’autonomie se décline au travers des notions d’action morale et de libre arbitre. En d’autres termes l’autonomie consiste à vouloir la loi et à s’y soumettre. Si maintenant on s’intéresse à sa vision en psychologie comportementaliste, Skinner dans [208] avance que l’autonomie et le libre arbitre ne sont qu’une illusion car c’est l’environnement qui détermine le comportement. Plus récemment Bouvet dans [38] pointe l’ambiguïté de définition de l’autonomie et lui préfère le terme anglo-saxon d’*empowerment* c’est à dire du pouvoir d’agir. Ces définitions non consensuelles, qui font référence au système biologique intelligent qu’est l’Homme, mettent donc en avant les notions de libre arbitre, de capacité d’action et l’importance que peut avoir l’environnement.

Venons en maintenant au domaine des systèmes artificiels et plus particulièrement de la robotique où le concept d’autonomie a été et sera de plus en plus étudié. La table 2.1 tirée de [230] et enrichie, en propose un ensemble de définitions. Même si elles ne relèvent pas du même niveau de granularité, elles mettent en avant que le concept d’autonomie est lié :

- À l’absence de supervision humaine [27] [58] [230].
- À la capacité qu’a le système à exécuter les actions demandées [230] [11].
- À la poursuite d’un objectif [11] [169].
- À la capacité d’adaptation du système au contexte interne [169] et externe [11] [58] [169] [219] d’exécution des actions.

Citation	Idées / Mots-clefs	Référence
<i>"Agents opérants sans intervention directe d'humains et ayant un certain contrôle sur leurs actions et états internes"</i>	<ul style="list-style-type: none"> <li>• Absence de supervision</li> <li>• Contrôle des actions</li> </ul>	[230]
<i>"Le robot doit être capable d'exécuter ses actions et de raffiner ou de modifier la tâche et son propre comportement en accord avec l'objectif courant et le contexte d'exécution de cette tâche"</i>	<ul style="list-style-type: none"> <li>• Exécution des actions</li> <li>• Adaptation</li> <li>• Objectif</li> <li>• Contexte</li> </ul>	[11]
<i>"Le robot peut s'adapter aux changements de son environnement, ou à ses propres changements (pannes) et continuer à poursuivre un objectif"</i>	<ul style="list-style-type: none"> <li>• Adaptation</li> <li>• Objectif</li> <li>• Contexte</li> </ul>	[169]
<i>"Capacité du robot à s'adapter à la variabilité de son environnement"</i>	<ul style="list-style-type: none"> <li>• Adaptation</li> <li>• Contexte</li> </ul>	[219]
<i>"Système capable d'opérer dans un environnement réel sans aucune forme de contrôle externe"</i>	<ul style="list-style-type: none"> <li>• Absence de supervision</li> <li>• Contexte</li> </ul>	[27]
<i>"Capacité à fonctionner indépendamment d'un opérateur humain ou d'une autre machine en exhibant des comportements non triviaux dans des environnements complexes et variables"</i>	<ul style="list-style-type: none"> <li>• Absence de supervision</li> <li>• Contexte</li> </ul>	[58]

TABLE 2.1 – Robotique : Quelques définitions de l'autonomie

L'absence de supervision et la capacité à exécuter les actions demandées semblent être deux dimensions évidentes pour un système autonome. Certains chercheurs tel que [68] vont encore plus loin en avançant qu'un système ne peut être réellement autonome que s'il est capable de se construire par lui-même. Sinon, il ne peut utiliser que les lois mises en place lors de sa conception ce qui limite sa capacité d'action et d'adaptation. Ce point de vue est d'une certaine façon corroboré par le principe de rationalité limitée de Simon [207] qui énonce que les actions d'un système sont nécessairement contraintes par ses capacités de calcul et par les informations et algorithmes dont-il dispose. Ce point de vue pourrait être cependant plus nuancé dans les prochaines années avec l'avènement des techniques récentes d'apprentissage de l'intelligence artificielle telles que le deep-learning [94].

La notion d'objectif, pourtant peu présente dans les définitions de la table 2.1 nous semble indissociable de la notion d'autonomie. En effet, un système, qu'il soit biologique ou artificiel, effectue toujours ses actions dans un but défini. La prescription de ce dernier reste cependant un sujet de débat philosophique autour de la notion d'agent autonome. Certains chercheurs tels que [151] avancent que des agents ne peuvent être réellement autonomes que s'ils sont capables de générer leurs propres buts. Ce point de vue est cependant trop absolu pour une application pratique à la robotique. En effet on peut constater que tous les systèmes, y compris ceux biologiques et en premier lieu l'Homme, ne déploient leurs activités que dans un cadre contraint par d'autres agents, et par des règles édictées (sociétales ou les 3 lois d'Asimov). En d'autres termes l'autonomie absolue nous semble être un concept qui relève, pour l'instant, plus de la philosophie que d'une préconisation pratique applicable à la robotique

Enfin la capacité d'adaptation d'un système artificiel face à la variabilité tant interne (tolérance aux fautes), qu'externe (environnement) est de toute évidence un des marqueurs importants de l'autonomie. Il est largement mis en avant dans les définitions de la table 2.1. Dans [89] les auteurs ajoutent qu'en robotique l'autonomie est aussi fortement liée à la diversité des tâches que le robot peut effectuer et des environnements dans lesquels il peut évoluer. En effet, de nos jours, malgré les progrès de la robotique et un degré de maturité certain des différentes fonctions devant être mises en œuvre pour doter les robots d'autonomie (localisation, planification de mouvement, déplacement, etc.), les missions robotiques autonomes restent rares. Il est possible de parvenir à un excellent niveau d'autonomie dans un environnement manufacturier parfaitement connu et instrumenté, présentant une faible dynamique, où les robots sont essentiellement mono-tâche. En revanche cet objectif s'avère nettement plus ardu en environnement extérieur, inconnu et changeant. C'est pourquoi plusieurs taxonomies du niveau d'automatisation d'un système artificiel ont été proposées dans la littérature [26] [178]. En fonction de l'implication de l'être Humain, le degré d'automatisation est alors décomposé en un ensemble variable de niveaux allant de la téléopération, à l'absence d'assistance humaine.

Les robots d'exploration extra-planétaire, qui restent très largement supervisés, sont un bon exemple de la difficulté à réaliser des missions totalement autonomes en environnement ouvert variable mais peu dynamique. Les rovers *Spirit* et *Opportunity* (2004) étaient téléopérés à très bas niveau (moteur) et ne parcouraient que quelques dizaines de mètres par jour. Depuis *Curiosity* (2012) a pu parcourir plus d'un kilomètre par jour mais en restant cependant téléopéré au niveau tâche [89]. En robotique terrestre, on ne peut évidemment passer sous silence la navigation autonome automobile au travers du projet DARPA [220] et du développement de la voiture sans chauffeur [77]. Si les succès sont indéniables dans cet environnement extrêmement dynamique, ils sont le fruit d'investissements colossaux, de la mobilisation d'équipes d'ingénieurs et de chercheurs pléthoriques. Ils nécessitent d'embarquer à bord du véhicule des batteries de capteurs extrêmement performants (et donc coûteux) ainsi qu'une puissance de calcul importante pour réaliser une tâche certes complexe mais très spécifique. Malgré cet in-



vestissement tant humain que matériel, les véhicules autonomes peinent encore à prendre en compte tous les scénarios de navigation envisageables. Sans compter que le niveau de sécurité visé doit être extrêmement élevé et que l'opinion publique n'est certainement pas prête à accepter qu'un véhicule "intelligent" soit à l'origine d'accidents mortels. Sous l'eau on peut aussi citer le développement d'AUV au long cours de type *glider* [163] qui peuvent rester des mois immergés pour collecter des informations. Cependant le contrôle de ce type d'engins sous-marins reste très limité, de même que la nature des tâches robotiques qu'ils sont à même d'exécuter.

La capacité d'adaptation d'un système biologique ou artificiel, qui s'inscrit comme LE marqueur de l'autonomie, dépend bien évidemment des capacités fonctionnelles de l'agent considéré. On distingue alors l'autonomie descriptive [39] qui se rapporte aux actions qu'il peut effectuer, de l'autonomie prescriptive qui correspond aux actions qu'il est en mesure d'effectuer. Mais c'est surtout l'aptitude à effectuer des choix en présence d'un problème à résoudre, c'est à dire de la capacité de décision d'un système qui lui confère une réelle liberté d'actions. Dans [245] les auteurs proposent de distinguer l'autonomie délibérative qui correspond à la capacité de décider le plan d'actions à suivre pour accomplir une tâche donnée, de l'autonomie décisionnelle qui correspond à la capacité de décider comment le faire, et pour finir de l'autonomie opérationnelle qui correspond à la capacité d'exécuter une tâche.

Plus récemment [109] propose une analyse très fouillée du concept de délibération en robotique. Pour les auteurs la délibération ne peut se réduire uniquement à la fonction de planification de tâches. L'action délibérée, c'est à dire issue d'un raisonnement motivé par des objectifs imposés, s'exprime au travers de six fonctions délibératives que sont la planification, l'action, l'observation, le monitoring (comparaison prévision / réalité), le raisonnement guidé par les buts et enfin l'apprentissage.

Pour notre part nous estimons qu'un système peut être considéré comme autonome si, pour des objectifs donnés, il est capable d'apporter lui-même des réponses aux questions suivantes pour les atteindre :

- Décider quoi faire (quelles tâches exécuter) ?
- Décider comment le faire (quelles ressources utiliser) ?
- Décider à quel moment le faire ?

Comme [41] nous décomposons l'autonomie selon deux points de vue (Figure 2.1). D'une part l'*autonomie décisionnelle* qui permettra, au plus haut niveau, de choisir, dans le temps, les tâches à exécuter pour répondre aux objectifs imposés. Cela correspond donc à la fonction de planification de tâches. D'autre part l'*autonomie comportementale*, qui à un instant donné, en fonction des tâches à exécuter, des ressources matérielles et logicielles et du contexte environnemental, permet de choisir celles qui permettront de satisfaire les objectifs fixés.

Enfin, la prise de décision nécessite d'avoir recours à un élément de choix permettant de déterminer, à un moment donné et parmi les alternatives possibles, celle permettant au mieux d'atteindre les objectifs. La *performance* visée du système s'impose d'elle même comme un critère pertinent pour effectuer cette sélection.

Nous avons rappelé dans ce paragraphe combien la notion d'autonomie était difficile à cerner. L'analyse réalisée a cependant mise en évidence l'importance que revêt la capacité de prise de décision pour un système autonome et son lien avec le concept de performance. Nous allons maintenant nous focaliser sur cette notion et sa déclinaison en robotique.

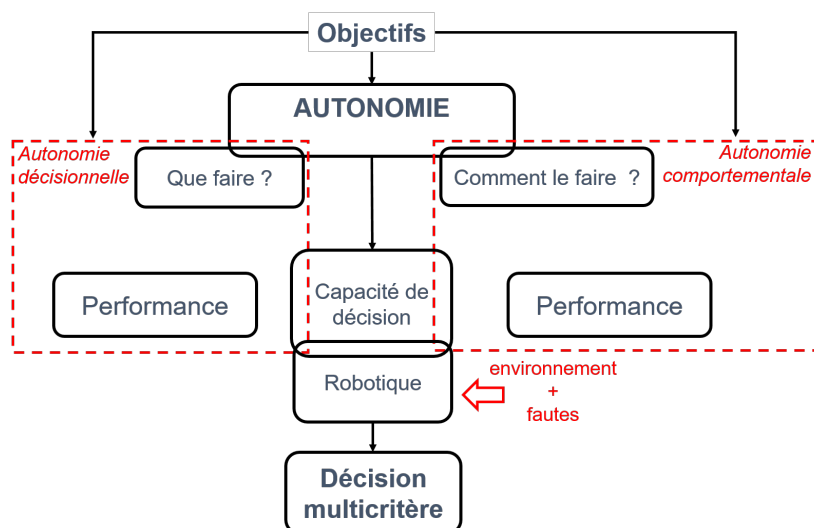


FIGURE 2.1 – Autonomie, décision et performance

## 2.2 De la performance à la garantie de performance : Positionnement général des travaux

Le concept de performance est largement utilisé dans de très nombreuses disciplines (mécanique, informatique, robotique, etc..). Cependant le sens avec lequel cette notion est employée reste le plus souvent flou et très dépendant des auteurs. Dans cette section nous allons donc tenter de mieux cerner ce concept et de préciser sous quelles formes il peut être rencontré en robotique en général et plus particulièrement dans les travaux que nous menons au sein de l'équipe EXPLORE.

### 2.2.1 Le concept de performance

Bourguignon, dans [37], caractérise la performance de la façon suivante :

- Elle est liée à un objectif.
- Elle peut avoir plusieurs dimensions.
- Elle est le résultat d'une action.
- Elle dépend des moyens mis en œuvre.
- Elle consiste à rapprocher le résultat d'une action, d'un objectif visé.

Ces différentes caractéristiques montrent que la performance est essentiellement caractérisée par 3 paramètres que sont les *moyens*, les *objectifs* et les *résultats*. En effet elle est étroitement liée à l'interprétation du résultat d'une action du système étudié, estimé ou réel, vis-à-vis d'un objectif fixé. Dans le premier cas, comme [213] on parlera d'*évaluation* de performance *à priori* et dans le second d'évaluation de performance *à posteriori*. Enfin, évidemment le résultat mesuré dépendra fortement des ressources (moyens) mobilisées.

Dans [32] et [157] les auteurs ajoutent un 4<sup>ème</sup> paramètre pour caractériser la performance, la *finalité*. Cela les conduit à proposer d'analyser la performance au travers de 4 prismes complémentaires :

- La pertinence : articulation entre moyens et objectifs.
- L'efficience : articulation entre moyens et résultats.
- L'efficacité : articulation entre objectifs et résultats.
- L'effectivité : articulation entre moyens, objectifs et résultats.

Le prisme de l'efficacité est évidemment celui qui est le plus souvent retenu par les auteurs lorsqu'ils parlent de performance. Dans nos travaux nous y ajouterons celui de l'efficience. La pertinence est une question évidemment essentielle qui doit être posée dès le début de la réflexion. En effet, il est illusoire de vouloir atteindre des objectifs qui sont hors d'atteinte compte tenu des moyens qui peuvent être mobilisés. D'autre part, il est évident que le surdimensionnement des moyens conduit de fait à minimiser le critère d'efficience et à maximiser celui d'efficacité ... mais à quel prix ! Pour finir l'effectivité donne un point de vue nous semble-t-il plus global car englobant les autres dimensions de la performance.

## 2.2.2 Le pilotage de la performance

Il est possible de piloter un système de façon à ce qu'il maintienne le niveau de performance attendu. Pour cela il est indispensable de disposer d'*indicateurs de performance*.

Un indicateur de performance est une donnée quantifiée permettant de mesurer l'efficacité de tout ou partie d'un système réel ou simulé par rapport à une norme, un plan ou un objectif [87]. On distingue les *indicateurs de résultat* qui renseignent sur la réalisation d'un objectif mais ne permettent pas d'agir sur le système. Ce sont uniquement des observateurs dénués de toute capacité d'action. A contrario, les *inducteurs de performance* établissent un lien causal entre l'objet de la mesure et la performance. Ils permettent donc de piloter le système. La construction des classiques structures causes-effets arborescentes permet d'identifier les indicateurs de résultat et les inducteurs de performance [158].

On peut alors construire un modèle de centre de pilotage d'un système [212] conformément à la Figure 2.2. Elle montre l'importance des indicateurs de performance dans le choix des actions à mettre en œuvre pour maintenir le niveau d'efficacité recherché d'un système piloté.

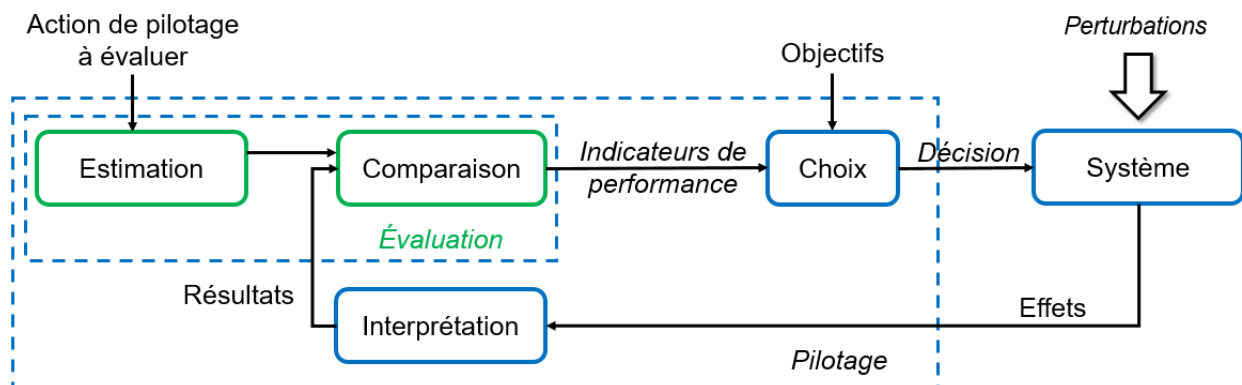


FIGURE 2.2 – Pilotage et performance

Le concept de performance venant d'être cerné dans sa formulation la plus générale nous allons maintenant balayer les formes qu'il peut prendre dans le domaine de la robotique puis plus particulièrement comment nous l'avons caractérisé au sein de l'équipe EXPLORE.

## 2.2.3 Performance et Robotique

Le concept de performance est couramment employé en robotique, car, comme le soulignent les travaux du comité technique TC-PEBRAS (*Technical Committee on Performance Evaluation and Benchmarking of Robotics and Automation Systems*) [106] de l'IEEE, il est maintenant indispensable de pouvoir quantifier les performances des systèmes robotisés au travers de métriques et méthodes d'évaluation bien définies. Cependant ce travail n'est pas arrivé au même degré de maturité en fonction des domaines abordés en robotique.

### 2.2.3.1 Robotique industrielle

Par exemple si l'on s'intéresse dans un premier temps à la robotique manufacturière, la dimension industrielle a conduit à la proposition d'un ensemble précis et accepté de critères de performance par les instances de normalisation [8] [191] tels que la précision du positionnement statique, le temps de cycle ou la répétitivité point-à-point, etc... Sans oublier la dimension sécuritaire qui ne peut être ignorée dans un environnement industriel et est abordée dans les normes [3] [190] où l'on définit, entre autres, une force, une puissance et une vitesse limite pour les robots évoluant en présence d'un être humain. Ces travaux ont été complétés par des critères plus spécifiques comme la dextérité [67] pouvant être aussi considérée pour les robots médicaux ou les capacités dynamiques des robots [232] en présence de vibrations. On doit cependant souligner que l'univers industriel relève très majoritairement de robots manipulateurs travaillant dans un environnement limité, souvent connu, pas ou faiblement dynamique, en l'absence de contrainte énergétique, où le robot pourra se localiser avec facilité. Même si les contraintes de précision, de répétabilité ou de rapidité d'exécution peuvent être très importantes et difficiles à satisfaire, les tâches à exécuter restent le plus souvent assez simples, sans beaucoup de variabilité et ne nécessitent que de faibles capacités décisionnelles. Ainsi les capacités délibératives de ces robots étant limitées les performances robotiques évaluées relèvent plus de l'automatisation que de l'autonomie.

### 2.2.3.2 Robotique mobile

A contrario, la robotique mobile autonome s'inscrit le plus souvent dans un contexte très différent de la robotique industrielle. En premier lieu le robot va se voir confier une mission, qui en fonction de sa complexité, sera déclinée en une succession de tâches robotiques devant être exécutées séquentiellement et/ou simultanément. On ne parle pas de mission pour un robot industriel. Par ailleurs, la variabilité de l'environnement au travers de sa dynamique et/ou de ses caractéristiques ((a)symétrie, propriétés physiques, etc..) s'inscrit très souvent comme une différence très marquée avec le milieu industriel. Enfin, un robot autonome n'est pas relié directement au réseau électrique et, par conséquent, la gestion de la dimension énergétique devient alors une problématique centrale de l'autonomie qui est totalement absente pour les robots industriels.

Ce cadre d'application étant moins standardisé et normatif que celui de l'industrie, un ensemble de critères de performance a vu le jour, de façon assez dispersée, au gré des centres d'intérêts des chercheurs et des fonctions robotiques qu'ils cherchaient à caractériser. On dénombre tout d'abord un corpus de travaux s'intéressant plus particulièrement à la performance d'une tâche robotique donnée. Cette décomposition orientée tâche se retrouve clairement dans le rapport [33] où les auteurs identifient à partir d'une classification des tâches de la robotique mobile (Simultaneous Localization And Mapping (SLAM), contrôle du déplacement, évitement d'obstacle, asservissement visuel, préhension, autonomie/tâches cognitives), les critères de performance pouvant s'y rapporter. Curieusement la dimension énergétique est absente de cette étude et toutes les classes identifiées ne relèvent pas du même niveau d'abstraction. De façon plus spécifique l'analyse consacrée aux plateformes domestiques intelligentes [34] identifie des critères de performance liés à la localisation (précision de la localisation), à l'énergie (énergie maximale embarquée, consommation à l'arrêt, durée d'utilisation), à la sécurité (évitement d'obstacle, de chute). Les articles qui viennent d'être cités se contentent d'énumérer des tâches robotiques et les critères associés. D'autres évidemment évaluent spécifiquement la performance d'une tâche robotique de base. Par exemple [135] analyse la performance de préhension d'une main sous-actionnée en termes de magnitude des forces de contact et en capacité de manipula-

tion. Pour l'évitement d'obstacle dans [193] les auteurs caractérisent d'une part l'environnement d'évolution en termes de densité (surface occupée/libre), de confinement, d'encombrement, etc..., et d'autre part la trajectoire suivie en fonction de son optimalité et de sa sécurité. Dans [49] les auteurs s'intéressent à la seule tâche de déplacement entre 2 points pour laquelle ils choisissent des critères de performance classiques comme la durée de déplacement, l'erreur finale de positionnement, la longueur du chemin, la distance aux obstacles, la courbure de la trajectoire, ainsi que les erreurs de tracking et longitudinales. Bien évidemment des auteurs font aussi appel au concept de performance pour qualifier des méthodes de localisation comme dans [164] où la précision de localisation d'un robot omnidirectionnel par filtre particulaire est analysée au regard du placement d'un système de radio-identification. Dans [129] la qualité de localisation pour des systèmes à base de capteurs laser est étudiée, en environnement dynamique, en s'intéressant à l'orientation estimée du robot et la robustesse des mesures effectuées. Dans [139] les auteurs comparent plusieurs méthodes de SLAM en choisissant une métrique non pas basée sur la qualité de la carte elle-même mais sur la localisation du robot au sein de celle-ci.

Nombre de ces critères de performance se retrouvent dans la liste dressée dans [53] pour évaluer la qualité de navigation d'un robot mobile. On y trouve pêle-mêle des critères variés relevant de la sécurité (nombre de collisions observées -mission, par unité de distance, par unité de temps- la distance aux obstacles -minimum, médiane), la longueur du chemin exécuté, la durée de la mission, la qualité de la trajectoire. Les organismes de normalisation se sont emparés plus récemment du domaine de la robotique mobile avec la proposition de critères de performance, tels que la précision de la localisation, la répétabilité, de la capacité de détection et d'évitement d'obstacle, pour la navigation [5], ou tels que la vitesse nominale, les conditions d'arrêt, l'angle maximal et la vitesse associée sur un plan incliné, la largeur de giration, pour le déplacement de robot à roues en environnement intérieur [4]. Des travaux récents [104] [103] s'appuient sur la norme ISO 25010 [7] relative aux modèles de qualité du système et du logiciel pour décomposer les objectifs d'une mission selon leur caractère fonctionnel ou non. Dans les premiers on retrouve ceux orientés tâches robotiques (navigation, perception, etc.), alors que les seconds sont orientés utilisateur et sont décomposés en performance mission (durée, qualité de service), énergie et autres (sécurité, robustesse, etc.).

Au confluent de la robotique industrielle et de la robotique mobile on trouve les travaux sur les robots manipulateurs placés sur des bases mobiles et utilisés dans un environnement industriel pour déplacer du matériel de poste de travail en poste de travail. Il semble d'après [36] que les travaux sur la performance de mesures effectuées dans un tel contexte soient inexistantes. Des critères de performance relevant tant de l'univers de la manipulation que du déplacement d'une base mobile sont donc proposés.

### 2.2.3.3 Mission robotique et performance

D'autres études s'inscrivent dans une perspective plus globale en s'intéressant à des performances qui relèvent plus de l'attente qu'a l'utilisateur vis-à-vis du système robotique pour exécuter des tâches complexes. Elles se focalisent donc sur des missions qui nécessitent la mobilisation de la plupart des tâches robotiques basiques que nous avons évoquées plus haut. On peut citer par exemple les travaux autour de la problématique de couverture robotique d'une surface à explorer (*Coverage task*) à l'aide d'un [73] [229] ou plusieurs robots [238]. Dans ces articles les auteurs cherchent à identifier des critères permettant de caractériser l'efficacité des algorithmes utilisés (pourcentage de couverture, longueur du chemin, énergie déployée) voire à proposer des critères de garantie de couverture. Ces études trouvent un prolongement dans les travaux effectués dans le cadre de la recherche et du sauvetage (*Search and Rescue*) tels

que [18] [153] qui caractérisent la performance des algorithmes d'analyse d'images utilisés pour localiser des victimes.

Un point de vue encore plus général est adopté dans le travail présenté dans [198]. Les auteurs considèrent, à juste titre, que la performance d'un système robotique devant réaliser des tâches complexes peut se décomposer au travers d'un ensemble de tâches plus simples ayant leurs propres critères de performance. Ils avancent qu'il existe évidemment des interdépendances fonctionnelles entre les performances obtenues globalement par le système, et celles que celui-ci peut atteindre au niveaux des tâches plus élémentaires. Ils proposent d'utiliser une technique d'apprentissage pour définir un modèle probabiliste [197] permettant de relier les différentes métriques de performance et d'optimiser le paramétrage des tâches simples pour satisfaire des contraintes de performance globales.

Au regard de ce tour d'horizon de l'emploi de la notion de performance en robotique on peut donc constater que le degré de maturité de ce concept dépend fortement de son domaine d'utilisation. La robotique industrielle l'a depuis longtemps identifié et parfaitement codifié alors que son utilisation en robotique mobile autonome reste très largement guidée par les besoins utilisateurs et qu'aucune réflexion avancée ne semble être actuellement menée au confluent de ces deux univers.

Par ailleurs on peut constater que le chercheur et/ou l'utilisateur, quelque soit le domaine considéré, ont besoin de disposer de métriques leur permettant d'évaluer et de comparer le niveau d'efficacité et donc la performance d'un algorithme donné, d'une stratégie, d'une tâche ou d'une mission robotique en s'intéressant à des critères pertinents, le plus souvent mesurés à l'issue d'expérimentations reproductibles (simulation, benchmarks). Enfin en robotique mobile, bien que de nombreux critères soient proposés, ceux relatifs à la sécurité, à la durée de la mission ou à la localisation semblent souvent être ceux invoqués par les chercheurs, même si le foisonnement de définitions différentes et l'absence de standardisation reste très présente par rapport à la robotique industrielle.

## 2.2.4 Notre approche de la performance

Le point de vue adopté dans le cadre de ce manuscrit pour définir le concept de performance relève directement des travaux menés au sein de l'équipe EXPLORE et détaillés dans [112] ou résumés dans [115] et [141].

Nous nous intéressons donc, dans le cadre de cette thèse, à la performance de missions robotiques autonomes. Si la performance est unique, en revanche ses points de vue sont multiples. Chacun de ces points de vue correspond à un *axe de performance*. Cependant la portée d'un axe dépend de sa spécificité, c'est pourquoi nous proposons de distinguer les *axes principaux* qui rassemblent tous les axes qui doivent être pris en considération quelle que soit la mission robotique envisagée, des *axes utilisateurs* qui eux relèvent d'un centre d'intérêt ne se retrouvant pas nécessairement au cœur de toutes les problématiques d'une mission.

Ainsi nous déclinons les axes principaux selon 5 dimensions :

- **La Sécurité** : l'aspect sécuritaire est évidemment indissociable des missions autonomes puisque le robot doit être capable de se déplacer sans danger pour lui même ou son environnement. Il se traduit usuellement au travers de la mise en place de capacité d'évitement d'obstacles tout en assurant l'innocuité des chocs inévitables en limitant l'énergie dissipée.
- **La Localisation** : cette dimension qui constitue le thème principal de ce document est centrale pour l'exécution de missions robotiques. Que le robot connaisse au préalable la carte de son environnement, ou qu'il la construise lors de son exploration, la tâche

robotique de navigation qui permet au robot de planifier son déplacement demande à ce que ce dernier soit capable de se localiser.

- **L'Énergie** : c'est un aspect bien souvent négligé dans le cadre des missions autonomes. Pourtant l'autonomie impose au robot de gérer l'énergie dont-il dispose de façon à pouvoir exécuter sa mission et, la plupart du temps, à pouvoir revenir à son point de départ, ou, pour le moins à un endroit où il est possible de le récupérer.
- **La Durée** : sur ce point, la réflexion de l'équipe a évolué depuis [112] qui positionnait la durée d'une mission parmi les axes utilisateurs. En effet, supposons que l'on demande au robot d'exécuter une mission en sécurité, en se localisant parfaitement, et en minimisant l'énergie consommée. Dès lors, si l'on ne fixe aucune contrainte concernant la durée de la mission, le robot n'a qu'à rester à sa position initiale, sans bouger, et tous les critères imposés sont satisfaits. Cela démontre clairement que la durée de la mission est un axe principal qui doit être défini quelque soit la mission autonome envisagée.
- **La Stabilité** : Cette dernière dimension peut tout d'abord être envisagée selon un point de vue mécanique. En effet il faut que lors de son déplacement le robot soit capable de prévenir tout risque de renversement qui mettrait évidemment une fin prématurée à sa mission. Mais elle peut aussi être reliée au regard de l'automaticien qui doit assurer, tout au long de la mission, que la boucle de contrôle converge. Cette dimension est fondamentale car elle permet de préserver le contrôle du système et d'en maintenir la réactivité. La fréquence de contrôle a évidemment un impact réel sur la dimension énergétique de la mission et son choix reste un problème ouvert pour les problèmes échantillonnés non-linéaires. Dans la suite de ce manuscrit nous supposerons que la fréquence de contrôle choisie permet d'assurer la stabilité du contrôle du système robotisé.

À ces axes principaux viennent s'ajouter, en fonction des besoins utilisateur d'autres dimensions de la performance plus spécifiques à la mission que l'on souhaite réaliser. À titre d'exemples pour ces axes utilisateurs, l'efficacité de couverture d'une zone, l'analyse de points d'intérêts explorés ou l'identification de ces derniers peuvent par exemple constituer autant de points de vue relevant plus spécifiquement de la *Qualité de Service* attendue pour le système robotisé.

Un axe de performance, quel qu'il soit, peut être caractérisé par un ensemble de paramètres :

- **Inducteur de performance** : Les inducteurs de performance permettent d'influencer la performance. Ils peuvent être non contrôlables, contrôlables hors ligne comme par exemple la quantité initiale d'énergie disponible, ou en ligne comme la vitesse du robot, le choix d'un capteur ou d'un algorithme particulier.
- **Indicateur de performance** : Un ou plusieurs indicateurs de performance permettent de mesurer l'efficacité du système étudié vis-à-vis d'un axe donné. Par exemple l'axe sécurité est évalué au travers de plusieurs indicateurs : la capacité d'évitement d'obstacle et l'innocuité en cas de choc.
- **Unité** : Un indicateur de performance peut avoir une unité, comme par exemple l'innocuité de l'axe Sécurité ou l'énergie qui peut se mesurer en Joule ou en Watt.heure.
- **Objectif (de performance)** : Un objectif de performance constitue la cible à laquelle on se rapporte pour évaluer la valeur mesurée d'un indicateur de performance.
- **Évaluation** : Chaque mesure d'un indicateur de performance doit être comparée par rapport à un objectif pour évaluer l'efficacité du système robotique selon l'axe de performance étudié pour l'indicateur choisi.

Pour terminer on peut distinguer différentes classes d'objectifs et de performance pour une mission en fonction des moyens et de l'instant de mesure (Figure 2.3). Considérons tout d'abord un axe de performance *axe* et un indicateur de performance *ind* de cet axe.

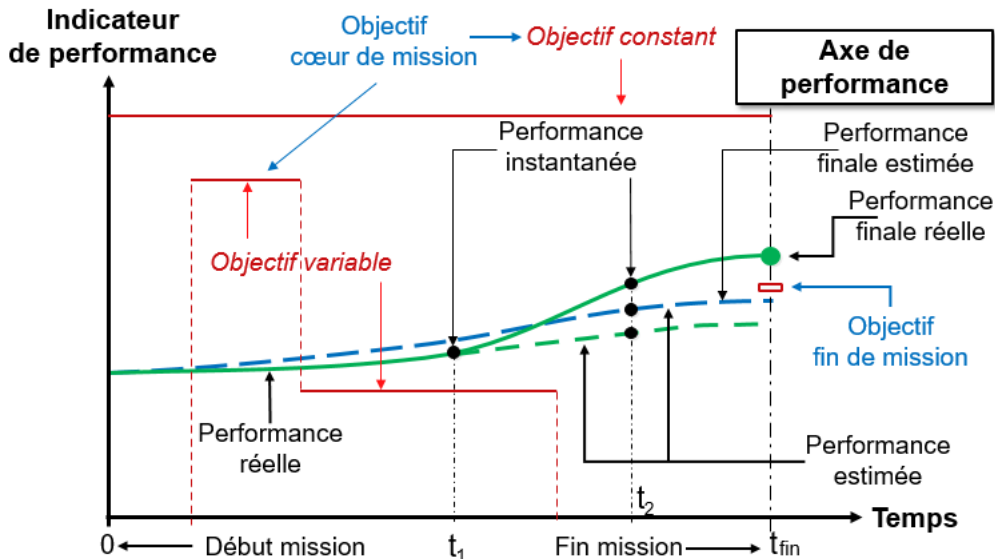


FIGURE 2.3 – Indicateurs et objectifs de performance

La valeur (la mesure) instantanée de cet indicateur, selon l'axe considéré à l'instant  $t$  est notée  $Perf_{ind}^{axe}(t)$  et correspond à une *performance instantanée* de la Figure 2.3. Cette valeur peut être une *performance estimée* (ligne en pointillés) lorsque l'on projette dans le futur l'évolution supposée du système robotique et que l'on utilise des modèles comportementaux (nécessairement approchés de l'indicateur concerné). Cette estimation peut être mise à jour (réévaluée), au fur et à mesure du déroulement d'une mission, dès qu'il est possible, à un instant donné, de mesurer la performance instantanée réelle. Cette *performance réelle* est représentée par un trait plein sur la Figure 2.3. La *performance finale*, qu'elle soit estimée ou réelle, est celle mesurée pour  $t = t_{fin}$ .

Si l'on s'intéresse maintenant aux objectifs il est aussi possible d'en différencier plusieurs catégories. Il existe tout d'abord des *objectifs de fin de mission* qui ne s'appliquent que pour  $t = t_{fin}$ . C'est par exemple le cas de la durée maximale autorisée ou de l'énergie maximale qui peut être consommée pour une mission. Mais il existe aussi des *objectifs cœur de mission* qui eux pourront évoluer ou rester constant durant le déroulement de la mission. Par exemple, si la sécurité doit toujours être satisfaite, les exigences de localisation (précision) peuvent varier contextuellement au cours d'une mission.

Maintenant que le concept de performance a été précisé dans le cas de la robotique mobile, nous allons nous intéresser à ce que recouvre la notion de garantie de performance qui constitue le cadre général de notre travail.

## 2.2.5 Garantie de performance - Cadre général de l'étude

Chercher à garantir la performance d'un système robotique autonome doit permettre de répondre à la question "*Le robot considéré est-il à même de remplir, sans intervention humaine, la mission qui lui est assignée, tout en satisfaisant l'ensemble des contraintes qui lui sont imposées ?*". Cet objectif dans l'absolu est inatteignable car qui peut prétendre qu'il est possible d'anticiper ou de pouvoir réagir à tous les événements prévus ou inopinés qui peuvent affecter le déroulement du service espéré ?

Cette question de garantie de service reste cependant pertinente car on doit être capable d'assurer à l'utilisateur, qu'en l'absence de circonstances trop défavorables (conditions environnementales nominales), le système autonome employé sera à même d'exécuter la mission envisagée.



Nous allons maintenant balayer certains travaux cherchant à fournir des garanties de performance de façon plus ou moins ciblée.

### 2.2.5.1 Garantie mono-axe (localisation)

Dans [179] [180] un modèle de contrôle prédictif stochastique est présenté pour garantir la stabilité pour des robots unicycles. La stabilité de ce modèle dépend d'une fonction de distance non-holonomique qui peut être représentée à l'aide d'une fonction de Lyapunov. La synthèse d'un contrôleur à garantie de performance est proposée dans [236] où une fonction caractérise le taux de convergence, le dépassement maximal et l'erreur statique. Cependant ces travaux ne prennent pas en compte l'impact que peut avoir l'échantillonnage du contrôle sur la stabilité du système étudié.

Des chercheurs se sont aussi intéressés à l'erreur de localisation et aux garanties qu'il est possible d'apporter. Volpe étudie et évalue expérimentalement dans [225] l'erreur de localisation d'un robot martien (Rocky 7) censé se déplacer en ligne droite. La position du véhicule est calculée en combinant une mesure odométrique du déplacement avec une estimation du cap obtenue à partir d'un compas solaire (référence absolue) ou d'un gyroscope. Il démontre qu'en fonction de la distance parcourue  $d$ , l'erreur de localisation croît théoriquement linéairement dans le 1<sup>er</sup> cas. Dans le second cas elle évolue quadratiquement avec  $d$  sur de petites distances ( $t \rightarrow 0$ ), et linéairement avec  $d$  sur de grandes distances ( $t \rightarrow \infty$ ).

Dans [174] les auteurs étudient par simulation le comportement de l'erreur de localisation engendrée par l'utilisation d'une technique d'odométrie visuelle sur de longues distances. Celle-ci croît asymptotiquement en  $o(d^{3/2})$ , où  $d$  représente la distance parcourue par le robot. Ils soulignent l'impact prépondérant de l'erreur d'orientation dans l'évolution super-linéaire de l'erreur de localisation. Plusieurs études confirment l'impact néfaste de l'absence d'orientation absolue sur l'erreur de localisation [23] [172].

Ces deux études ne donnent pas rigoureusement de borne, et donc de garantie sur l'erreur de localisation en raison de la dérive non bornée de la méthode employée. En revanche, pour une distance parcourue donnée il est possible de connaître une estimation de l'erreur de localisation puisque l'on dispose, en quelque sorte, d'un modèle prédictif. Enfin dans [133] une analyse à la fois théorique, en simulation et expérimentale (Pioneer P3Dx) du comportement de l'estimation de l'erreur de position obtenue à partir de mesures bruitées de la pose d'un robot est proposée. Expérimentalement les capteurs utilisés sont des odomètres pour l'estimation de la distance parcourue et une caméra pour l'estimation de la rotation relative et du vecteur de déplacement. Les auteurs se sont intéressés à l'évolution de la moyenne et de la covariance de l'erreur de position. Ils démontrent que le biais (moyenne de la différence entre la position réelle et celle estimée) et la variance de l'erreur d'estimation de position croissent asymptotiquement au plus linéairement avec le temps et la distance parcourue. Des formules exactes sont données pour des trajectoires rectilignes et périodiques. Un des résultats intéressants est que les auteurs prouvent que le vecteur de déplacement du robot (position courante par rapport au point de départ) est un élément clef de la détermination du biais. Ce paramètre est souvent confondu avec la distance parcourue puisqu'il se confond avec cette dernière dans le cas de trajectoires rectilignes. On peut cependant regretter que les validations expérimentales aient été effectuées dans des environnements de dimensions réduites et seulement pour des trajectoires circulaires.

Une analyse théorique très fouillée de la propagation des erreurs odométriques est présentée dans [13]. Une description analytique précise sous la forme de systèmes d'équations différentielles non linéaires est proposée. Celles-ci ne pouvant être résolues dans le cas général, une linéarisation est alors utilisée pour trouver une solution analytique approchée dans le cas de perturbations

systématiques et aléatoires quels que soient les capteurs et trajectoires considérés. L'auteur démontre évidemment que l'erreur odométrique dépend de la distance parcourue, mais aussi qu'elle peut être annulée si le robot revient à son point de départ en suivant le même chemin qu'à l'aller. Une analyse détaillée des moments d'erreurs (par analogie avec les moments utilisés en mécanique ou statistique) et de leurs comportements est effectuée afin d'identifier des propriétés dans le cas de trajectoires spécifiques (symétrique, fermée, réversible, ligne droite). On peut cependant regretter que ces résultats ne soient corroborés que par des simulations.

Enfin l'analyse par intervalle permet d'approcher de manière pertinente la question de garantie de localisation. Dans ce cadre on peut citer les travaux [99] [97] qui proposent une méthode ensembliste déterministe de localisation globale au sein d'une grille d'occupation (carte) expérimentée dans le cadre du défi CAROTTE (CARTographie par RObot d'un TErritoire) avec un robot MiniRex. Les auteurs supposent que toutes les mesures capteurs présentent des erreurs bornées de façon à pouvoir garantir un intervalle contenant la valeur recherchée et cherchent à déterminer à tout instant une boîte contenant de façon garantie la localisation du robot. Sa détermination s'appuie sur un algorithme (IAL - Interval Analysis Localization) décomposé en une étape de prédiction et une étape de correction, où le problème est formulé comme un problème de satisfaction de contraintes. Ces dernières prennent en compte des formulations de la distance robot-obstacle, de la distance entre 2 mesures et évidemment de la correspondance mesure-obstacle pour se positionner dans la carte de l'environnement. Les expériences ont été menées avec succès dans des environnements symétriques ou non. Le temps d'exécution reste faible sauf lorsque le robot doit se relocaliser à partir d'une situation inconnue (*kidnapping*). Ce type d'approche semble donc performant mais il définit uniquement une qualification instantanée de la garantie de localisation (ce qui est déjà beaucoup) sans proposer de modèle prédictif de son évolution. Des travaux récents extrêmement prometteurs ont été proposés dans [195] [196] sur le calcul de trajectoires garanties en proposant une nouvelle approche ensembliste d'intégration des équations d'états où le problème est formalisé comme un réseau de contraintes appliquées à un ensemble de trajectoires. À partir des équations du système et d'une estimation des incertitudes sur la loi de commande et sur la position initiale du robot, des tubes contenant l'ensemble des trajectoires envisageables sont construits. Des informations extéroceptives permettent de limiter la dérive induite par la navigation à l'estime. Les auteurs proposent aussi une approche de résolution intervalliste du problème de détection de boucles au sein de la trajectoire d'un robot [194]. Un nouvel algorithme de SLAM est aussi proposé s'appuyant sur l'hypothèse de la détection de mesures identiques inter-temporelles. L'ensemble de ces travaux est illustré sur des exemples représentatifs de la robotique ou sur des expérimentations de grande envergure de robotique sous-marine. Cependant ils ne s'inscrivent pas dans une logique prédictive mais plutôt dans objectif d'analyse permettant la reconstruction des trajectoires réelles à partir de données expérimentales spatialement et temporellement incertaines où les recalages sont rares ou difficiles à réaliser comme dans le milieu sous-marin. Enfin les durées de post-traitement peuvent être importantes et l'évaluation de l'incertitude de localisation n'est pas le sujet central de ces études.

L'utilisation de filtre récursif avec estimation et prédiction a aussi été proposée pour un environnement connu dans [70] [71] où des données absolues imprécises provenant de capteurs à faible fréquence d'échantillonnage (GPS) et des informations issues de capteurs proprioceptifs (odométrie, gyromètre) à haute fréquence sont mémorisées dans des buffers de dimensions limitées. En utilisant le modèle de déplacement du véhicule, ces données sont ensuite fusionnées en faisant appel à l'algorithme d'inversion ensembliste RSIVIA (Robust Set Inverter via Interval Analysis) [118] qui tolère un taux de données erronées. L'ajout de contraintes issues de la connaissance de la carte permet finalement de définir une boîte garantissant la position et l'orientation d'un robot. Cette approche a démontré sa robustesse sur un véhicule circulant sur la voie publique. En l'absence de données absolues le déplacement à l'estime reste sujet au non

bornage de l'erreur de position. Là encore les approches par intervalles ne proposent qu'une garantie "instantanée" et ne fournissent pas de modèle prédictif. Une approche voisine faisant appel à une analyse par intervalle est proposée dans [65]. Les auteurs y proposent un algorithme (BEES - Bounded Error Estimation Scheme) intégrant une démarche de suivi de localisation multi-hypothèses permettant de gérer des données incohérentes et les incertitudes du modèle d'évolution du robot ainsi que celles du modèle de l'environnement. Cette approche est évaluée en intérieur, en simulation et expérimentalement, sur un enceinte de test réduite (10m \* 10m) et équipée de marqueurs pouvant être repérés par un robot équipé d'un goniomètre. La précision de localisation, pouvant atteindre quelques centimètres, est dépendante du temps de calcul alloué et du taux de données erronées.

### 2.2.5.2 Garantie multi-axes

Certains travaux cherchent à prendre en compte simultanément plusieurs axes de performance. Dans [187] les auteurs cherchent à réduire la consommation énergétique d'un robot planétaire devant évoluer sur un chemin prédéfini avec une marge d'erreur donnée, en ne mobilisant le système de perception que lorsqu'il est nécessaire. L'ordonnancement des capteurs est alors formulé comme un problème de croyance probabiliste à l'aide de Processus Décisionnels Markoviens. En supposant que l'énergie mobilisée sur toute la trajectoire par les capteurs  $E_C$  est du même ordre de grandeur que celle dépensée par les moteurs  $E_M$ , ils cherchent à minimiser l'énergie totale consommée  $E$ . Puisque le système de localisation doit être initialisé avant d'être opérationnel, périodiquement, le robot a le choix d'utiliser ou non le système de localisation en consommant ou non l'énergie nécessaire correspondante. En l'absence de localisation, le robot navigue à l'estime et s'écarte de son chemin nominal (dérive odométrique) pour y revenir une fois relocalisé. La recherche de la solution d'ordonnancement de la perception permettant au robot de rester dans un couloir prédéfini autour du chemin est effectuée en faisant appel à un algorithme glouton (solution sous-optimale) où à un algorithme de programmation dynamique. Les résultats expérimentaux, sur différents types de sol, font apparaître un gain énergétique pouvant atteindre près de 60% de l'énergie capteur. Dans ce travail seule une garantie de localisation probabiliste est considérée. Qui plus est il suppose que le trajet est sans obstacle ce qui peut être dangereux. Enfin les distances expérimentales considérées restent modestes. Par ailleurs l'hypothèse d'équité de consommation entre le système de perception et de déplacement reste arbitraire. La prise en compte d'un couloir de déplacement fait apparaître implicitement la notion de marge de localisation. De plus la modélisation intègre que, pour certains capteurs ou certaines techniques de localisation, un temps de 'préchauffage' doit être pris en compte avant que la fonction de localisation soit opérationnelle.

De façon encore plus générale une approche originale développée dans [104] et [103] propose de s'appuyer sur le machine learning (Q-learning, Deep Q-learning) pour reconfigurer dynamiquement les ressources tout en respectant les exigences non-fonctionnelles (performance, énergie, sécurité, etc.) d'une mission robotique. Trois types de missions sont étudiées (Q-table) tout d'abord en simulation :

- Relier des points de passage à l'aide d'un robot Pioneer et analyser l'impact de la fréquence de contrôle et de la vitesse du robot sur la durée, la sécurité (distance à obstacle) et l'énergie.
- Assurer un rôle de serveur vidéo et étudier l'influence sur l'énergie dépensée du nombre de demandes de visualisation, de la taille des données et de la fréquence de l'image.
- Être capable d'effectuer une analyse sémantique d'une scène (reconnaissance d'objets et de personnes) et évaluer l'impact de la complexité de la scène sur le temps d'exécution de l'algorithme de reconnaissance et donc sur l'énergie dépensée.

En reprenant la même approche (apprentissage en simulation) au niveau mission, les auteurs proposent d'utiliser à nouveau ces approches de machine learning pour élaborer un manager de mission auto-adaptatif qui, en fonction de l'état courant du système robotique, et de la disponibilité des ressources va prendre la décision (modèle d'adaptation) la plus adaptée pour qu'il puisse atteindre les objectifs de performance visés. Enfin en plus des managers locaux, un manager hiérarchisé multi-missions est présenté pour gérer la consommation énergétique du robot en fonction des priorités (statiques ou dynamiques) fixées par l'utilisateur. L'approche complète proposée est validée expérimentalement démontrant une grande adaptation à la variabilité de l'environnement et sa capacité à prendre en compte simultanément et à gérer dynamiquement plusieurs dimensions de la performance.

Ce travail très complet et prometteur, faisant appel à des techniques d'intelligence artificielle récentes, présente cependant, à notre avis, des limitations. Tout d'abord au niveau conceptuel il semble étonnant de distinguer performance et énergie qui n'en est qu'un point de vue particulier. Par ailleurs l'approche de gestion de ressources demande à être précisée. En effet, le robot lui même est considéré comme une ressource pouvant être employée pour telle ou telle mission, mais la modulation de performance n'est en fait obtenue que par la modulation des paramètres de contrôle étudiés (inducteurs de performance) et non pas sur le choix des ressources matérielles et logicielles embarquées elles mêmes. Enfin, bien évidemment, la pertinence des approches par apprentissage repose sur la masse des essais qui ont pu être réalisés au préalable. L'apprentissage est très certainement une approche prometteuse qui peut s'avérer très intéressante pour construire et améliorer les modèles utilisés, même s'il faut rester prudent avant de parler de garantie.

### 2.2.5.3 Garantie par analyse formelle

Enfin, certains travaux sur l'analyse formelle relèvent aussi de la garantie de performance. [150] et [205] proposent une méthode de modélisation et d'analyse de la faisabilité de missions robotiques complexes décrite à l'aide de réseaux de Petri. Pour ce faire les auteurs décrivent des modèles permettant de représenter une fonctionnalité robotique (Activité). Celle-ci, qui peut s'exécuter de différentes façons (alternative), consomme et produit des ressources matérielles et/ou logicielles. Ces dernières peuvent être dans un état inaccessible (en panne), libre ou occupé. Dès lors le déroulement de la mission est automatiquement traduit en un réseau de Petri global sur lequel on effectue une analyse du graphe des marquages à l'aide l'outil TINA [31]. Il est ainsi possible de déterminer si la mission peut être réalisable, d'analyser les raisons d'un blocage ou de détecter la présence de boucles infinies d'exécution. Il est clair que ce type d'analyse permet à l'utilisateur de s'interroger sur certaines limitations matérielles et logicielles de la mission qu'il compte exécuter et lui permet ainsi d'engager des actions préventives. On peut cependant remarquer qu'il faudrait encore mener plus loin l'analyse des ressources critiques et que, par exemple l'ajout des probabilités de pannes dans le graphe permettrait d'engager une analyse stochastique qui serait très pertinente. Même si ce type d'analyse ne fournit pas des garanties sur des objectifs quantitatifs en terme de performance, la construction complète du graphe d'états renforce la confiance que l'on peut avoir dans le succès de la mission.

La question de la garantie de performance est aussi prise en compte explicitement dans [154] [155]. Ces travaux présentent un outil d'analyse de performance d'une mission permettant, avant d'exécuter cette dernière, de garantir sa faisabilité. On construit tout d'abord, à partir d'une description conviviale de la mission réalisée dans l'environnement MissionLab [15], le contrôleur, (machine à état fini) correspondant qui est lui même traduit en langage PARS (Process Algebra for Robot Schemas)[156] pouvant être utilisé pour effectuer une vérification formelle. L'utilisateur définit alors un ensemble de contraintes devant être satisfaites lors de l'exécution de la mission. L'algorithme de vérification VIPARS (Verification In PARS) prend

alors en compte les incertitudes liées aux déplacements du robot, aux bruits des capteurs et à l’environnement pour déterminer, par model checking, si les contraintes définies sont satisfaites ou non. Les performances analysées sont exprimées par exemple en termes de succès de la mission (liveness), de sécurité ou de taux de satisfaction d’une contrainte (durée mission par exemple) en faisant appel à des processus stochastiques. L’analyse des causes d’échec, là encore, conduit l’utilisateur à proposer des solutions permettant de pallier les difficultés rencontrées. Ce travail, bien que très intéressant et original, présente actuellement de nombreuses limitations. Tout d’abord il ne permet qu’une analyse hors-ligne de la mission en s’appuyant sur des modèles toujours perfectibles. D’autre part il ne propose pas d’alternatives en cas de non satisfaction de contraintes. Enfin, il reste employé sur des missions assez simples et le problème de l’explosion combinatoire auxquelles font inévitablement face les méthodes de Model Checking n’est pas réellement abordé.

#### 2.2.5.4 Notre approche de la garantie de performance

L’approche de garantie de performance PANORAMA (Performance and AutoNOmy using Resources Allocation Management) que nous proposons dans l’équipe EXPLORE et qui est détaillée dans [112] vise à choisir, configurer et maintenir, tout au long d’une mission, les ressources matérielles et logicielles à mettre en œuvre, pour assurer le succès du service demandé tout en vérifiant les contraintes de performance imposées par l’utilisateur. Dans le travail présenté la prise en compte des axes sécurité, durée et énergie est largement détaillée.

Si l’on considère un axe de performance  $axe$  et l’un des ses indicateurs de performance  $ind$ , alors la contrainte de performance  $Cper_{axe}^{ind}$  définit une relation d’ordre entre la performance observée  $Perf_{axe}^{ind}$  à un instant  $t$  et un objectif visé  $Obj_{axe}^{ind}$  selon la relation suivante (2.1) :

$$Cper_{axe}^{ind}(t) = Perf_{axe}^{ind}(t) \text{ op } Obj_{axe}^{ind}(t) \text{ avec } op \in \{<, \leq, =, \neq, >, \geq\} \quad (2.1)$$

En revanche le choix des algorithmes de localisation pertinents au vue des contraintes liées à l’environnement définies par l’utilisateur est supposé satisfaisant, tout comme le choix de la fréquence de contrôle permettant d’assurer la stabilité de la commande du système robotisé. C’est cette méthodologie, qui servira de cadre à notre travail, que nous allons maintenant présenter.

## 2.3 La Méthodologie PANORAMA

La méthodologie PANORAMA vise à doter d’une autonomie comportementale un système robotique devant exécuter une mission  $M$  complexe (multi-tâches) avec un certain niveau de performance dans un environnement connu. Le robot doit donc avoir la capacité de décider tout au long de la mission, parmi toutes les **Alternatives d’Implémentation AI** de ressources matérielles et logicielles envisageables, celle qui doit être utilisée à un instant donné.

Cela nous conduit à proposer différentes définitions pour les termes utilisés.

### 2.3.1 Quelques définitions

Une **Mission**  $M$  (Équation (2.2)) peut se décrire comme l’exécution d’une séquence de  $n_{Obj}$  Objectifs :

$$M = \{Objectif\}_{Seq} = Obj_1 \circ \dots \circ Obj_{n_{Obj}} \quad (2.2)$$

où  $\circ$  désigne l’opérateur de séquentialité

Un **Objectif**,  $Obj_i$  (Équation (2.3)) correspond à l'exécution concurrente d'un ensemble  $n_T$  de tâches. Son domaine d'application peut être défini spatialement (sur une distance) ou temporellement (sur une durée).

$$Obj_i = \{Tâches\}_{Conc} = T_1 || \dots || T_{n_T} \quad (2.3)$$

où  $||$  désigne l'opérateur de concurrence

Une **Tâche** (Équation (2.4)) peut être mise en œuvre en faisant appel à différentes **Option d'Implémentation de Tâche OIT**. Une OIT correspond à la sélection et au paramétrage de ressources matérielles et logicielles du système robotique. Une seule OIT doit être sélectionnée pour réaliser une tâche robotique.

$$\mathcal{E}(OIT)_T(t) = \{OIT_1, \dots, OIT_{k_1}\}_T \quad (2.4)$$

où  $\mathcal{E}(OIT)_T(t)$  représente l'ensemble des OIT pouvant être exécutées à  $t$  pour réaliser la tâche  $T$ .

À un instant donné de la mission, l'ensemble des tâches devant être concurremment mises en œuvre sont exécutées en sélectionnant et en paramétrant une **Alternative d'Implémentation AI** (Équation (2.5)) Pour cela, il faut donc sélectionner une unique OIT pour chacune des tâches impliquées à l'instant considéré.

Soit  $\mathcal{T}(T)(t) = \{T_1, \dots, T_n\}$  l'ensemble des tâches devant être exécutées à l'instant  $t$  et où  $n$  représente le nombre de ces tâches.

Soit

$$\begin{aligned} \mathcal{E}(OIT)(t) &= \{\mathcal{E}(OIT)_{T_1}(t), \dots, \mathcal{E}(OIT)_{T_n}(t)\} \\ &= \{\{OIT_{1,1}, \dots, OIT_{1,k_1}\}, \dots, \{OIT_{n,1}, \dots, OIT_{n,k_n}\}\} \end{aligned}$$

l'ensemble de toutes les OIT pouvant être exécutées à l'instant  $t$  pour mettre en œuvre les objectifs de performance, où  $k_i = \text{card}(\mathcal{E}(OIT)_{T_i}(t))$  correspond au nombre d'OIT de la tâche  $i$  à l'instant  $t$ .

Alors  $\forall T_i \in \mathcal{T}(T)(t) \exists! Sel_i / 1 \leq i \leq \text{card}(\mathcal{E}(OIT)_{T_i}(t))$

$$AI(t) = \{OIT_{Sel1}, \dots, OIT_{Seln}\} \quad (2.5)$$

Où  $OIT_{Sel_i}$  correspondant à l'OIT sélectionnée pour la tâche  $T_i$ .

Il est alors possible de définir le nombre  $n_{AI}(t)$  (Équation (2.6)) d'Alternatives d'Implémentation AI devant être considérées à l'instant  $t$  comme :

$$n_{AI}(t) = \prod_{i=1}^n \text{card}(\mathcal{E}(OIT)_{T_i}(t)) \text{ avec } \mathcal{E}(OIT)_{T_i}(t) \in \mathcal{T}(T)(t) \quad (2.6)$$

La Figure 2.4 illustre les concepts évoqués plus haut. On considère que l'objectif  $Obj$  qui doit être réalisé à l'instant  $t$  est mis en œuvre par le biais de 3 tâches robotiques  $T_1$  à  $T_3$ . Chacune de celles-ci dispose respectivement de 4, 2 et 3 options d'implémentation de tâche OIT. Certaines d'entre-elles nécessitent de fixer la valeur de paramètres,  $k, l$  pour  $OIT_{1,2}$  et  $m$  pour  $OIT_{3,1}$ . Le décompte de la combinatoire fait apparaître qu'il existe donc  $4 * 2 * 3 = 24$  Alternatives d'Implémentation différentes envisageables à l'instant  $t$ .

Les hypothèses générales suivantes sont supposées vérifiées :

- La quantité initiale d'énergie embarquée dans les batteries est connue.
- La carte de l'environnement de la mission est connue.
- Le chemin nominal devant être parcouru durant la mission est défini.

La méthodologie PANORAMA peut synthétiquement être représentée par la Figure 2.5.

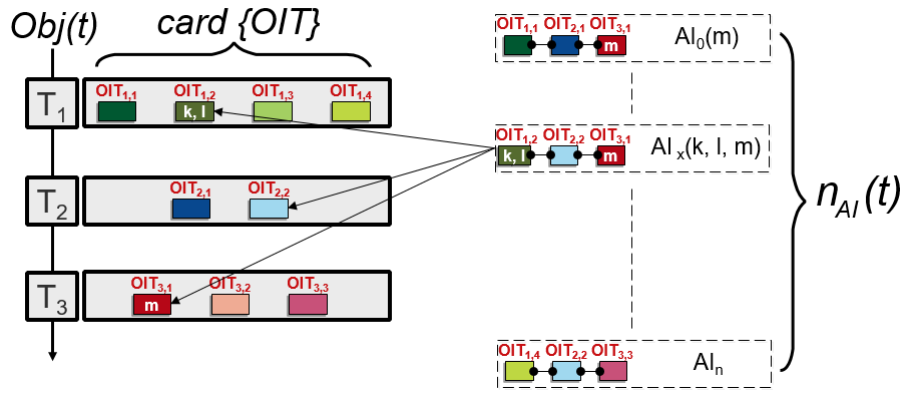


FIGURE 2.4 – Tâches - Option d'implémentation de Tâche (OIT) et Alternative d'Implémentation (AI) (d'après [112])

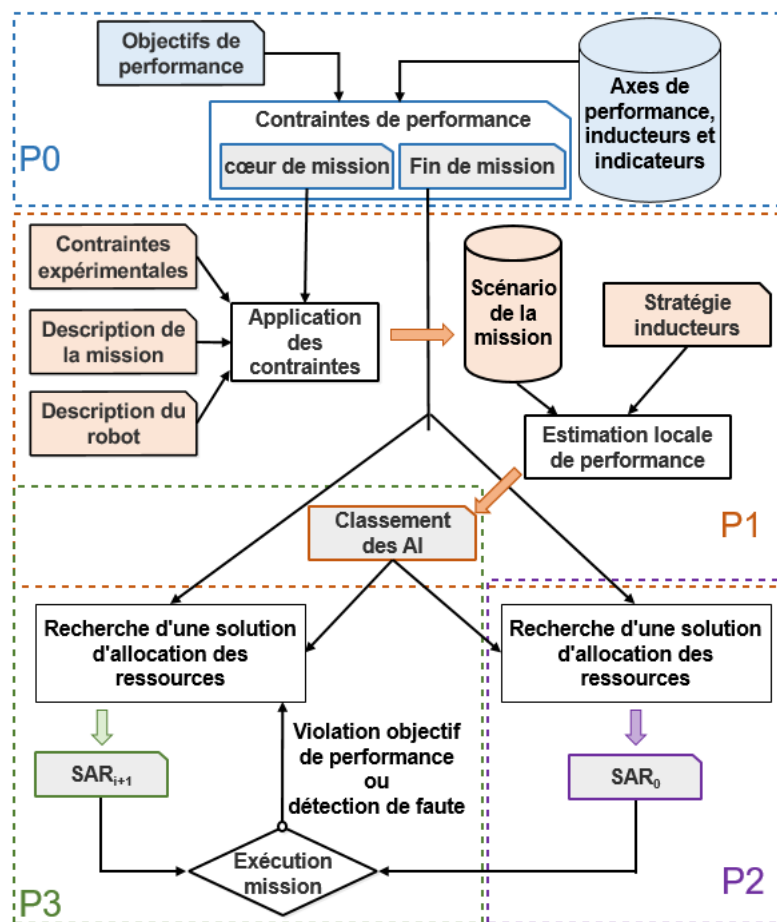


FIGURE 2.5 – La méthodologie PANORAMA

Elle peut se décomposer en 4 phases qui vont maintenant être détaillées.

### 2.3.2 P0 : Phase préparatoire

La phase initiale P0 est une étape assez longue et fastidieuse qui rassemble tout le travail préalable à la mise en œuvre de PANORAMA. Elle permet d'identifier les axes, indicateurs et inducteurs de performance pertinents de la mission ainsi que les objectifs qui seront visés. À partir de ces informations, elle met en évidence les contraintes qui devront être satisfaites durant la mission.

Évidemment, il faut en premier lieu s'interroger sur les axes de performances qu'il y a lieu d'étudier. En plus des axes des performance principaux : Sécurité, Localisation, Énergie et Durée et Stabilité, auxquels peuvent être ajouté des axes utilisateurs.

Pour chacun de ces axes il faut identifier les indicateurs de performance qui seront évalués tout au long de la mission et pour lesquels des objectifs de performance seront formulés. À un axe de performance peut correspondre plusieurs indicateurs de performance.

### 2.3.2.1 Identification des inducteurs de performance

Une fois les indicateurs de performance choisis, l'identification des inducteurs de performance permettant de les influencer est une étape particulièrement importante. Pour faciliter leur mise en évidence nous proposons de faire appel à un diagramme causes-effets qui permet d'identifier graphiquement les liens entre inducteurs et indicateur de performance.

Plusieurs classes d'inducteurs peuvent être identifiées en fonction des moyens de contrôle que nous pouvons avoir :

- **Non contrôlable** : Inducteur dont il est impossible de modifier les caractéristiques.
- **Contrôlable hors ligne** : Inducteur dont il n'est possible de modifier les caractéristiques qu'avant l'exécution de la mission.
- **Contrôlable en ligne** : Inducteur dont il est possible de modifier les caractéristiques durant l'exécution de la mission.

Il est possible d'identifier de nombreux inducteurs de performance dont la classification dépend souvent du point de vue selon lequel on se place.

- **Environnement** : L'environnement est bien évidemment un des paramètres qui influence grandement la performance du système robotique. Il peut être partiellement contrôlable hors ligne puisqu'on peut l'instrumenter (ajout d'amers géolocalisés par exemple) à sa convenance et parfois le modifier de façon à améliorer l'efficacité de certains algorithmes. Il faut aussi souligner l'impact qu'il peut avoir sur les mesures capteurs et les incertitudes associées en fonction de sa nature (absorption, matière, etc...). Sans compter qu'un environnement connu peut comporter une certaine dynamique avec l'ajout ou le déplacement d'éléments.
- **Ressources matérielles** : Toutes les ressources matérielles sont contrôlables et constituent donc des inducteurs de performance.
  - **Robot** : Le robot lui même est un inducteur de performance hors ligne. En effet le choix des caractéristiques physiques, géométriques, matérielles et logicielles constitue bien évidemment un facteur déterminant pour la réussite d'une mission donnée dans un environnement.
  - **Capteurs** : Les capteurs constituent autant d'inducteurs de performance en ligne s'il est possible de les (dé)connecter à loisir en fonction des besoins de la mission et de leurs caractéristiques (précision, consommation énergétique, etc...). Ils vont ainsi influencer les dimensions sécuritaire, énergétique et de localisation de la mission. Ils peuvent par ailleurs disposer de paramètres de reconfiguration (fréquence d'utilisation par exemple) pouvant être modifiés tout au long de la mission.
  - **Actionneurs** : Ils sont bien évidemment contrôlables en ligne puisqu'ils permettent d'exercer une action sur l'environnement. La vitesse de déplacement est bien évidemment un inducteur clef impactant la durée de la mission et la consommation énergétique.



- **Ressources logicielles** : Toutes les ressources logicielles sont aussi contrôlables et constituent donc aussi des inducteurs de performance.
  - **Algorithmes** : Évidemment le choix des algorithmes embarqués est crucial pour la réussite d'une mission et l'atteinte des objectifs de performance. Ils sont choisis hors ligne mais mobilisés en ligne lors de l'exécution de la mission. La sélection d'un algorithme donné va impacter différentes dimensions de la performance comme la charge du processeur et la mémoire utilisée, l'énergie consommée ou la qualité des données générées. Certains paramètres des algorithmes peuvent aussi constituer des inducteurs comme par exemple le chemin nominal à suivre par le robot. On conçoit bien que pour relier deux points il peut exister une infinité de trajectoires possibles. Dès lors la longueur du chemin choisi, sa courbure, et les zones traversées peuvent très fortement influencer certains indicateurs de performance tels que l'énergie et la durée par exemple.
  - **Architecture de contrôle** : Elle gère le déroulement de la mission et donc la sélection des ressources matérielles et logicielles à utiliser. Le choix du middleware embarqué impacte les capacités de reconfiguration ainsi que le déterminisme de l'ordonnancement des tâches. Cela constitue donc un inducteur hors ligne. En ligne, la fréquence de contrôle a un impact direct sur la stabilité du système (lois de commande) mais aussi sur sa réactivité (sécurité) et l'énergie consommée. Lorsque cela est possible, on peut aussi jouer sur la fréquence du processeur pour moduler sa consommation énergétique.
- **Ressources énergétiques** : On ne peut pas parler de mission autonome sans prendre en compte cet inducteur de performance. C'est un inducteur hors ligne grâce auquel on peut maximiser l'énergie initialement embarquée. Il faudra alors gérer ce capital, ou sa régénération lorsque cela est possible, tout au long de la mission de façon à pouvoir l'achever.

### 2.3.2.2 Formulation des modèles prédictifs d'estimation de la performance

Une fois les inducteurs de performance contrôlables en ligne identifiés, il faut être capable d'établir un modèle permettant de traduire comment chacun d'entre eux influence le ou les indicateurs auxquels ils sont reliés. On supposera que la loi permettant de composer l'influence de tous les inducteurs contrôlables sur un indicateur de performance a été déterminée.

En ce qui concerne l'axe Durée si l'on suppose que le robot se déplace à  $v_{rob} = cte$  et qu'il parcourt un chemin de longueur  $\Delta l$  alors la durée  $\Delta T$  de ce déplacement est donnée simplement par l'Équation 2.7 :

$$\Delta T = \frac{\Delta l}{v_{rob}} \quad (2.7)$$

Ainsi si l'on connaît la longueur  $L$  du chemin qui doit être suivi par le robot lors de sa mission et que l'on peut le décomposer un ensemble  $\mathcal{Z}1$  de zones  $z_{1,i}$  de longueur  $\Delta l_i$  où la vitesse  $v_{rob_i}$  sera constante, et en un ensemble de zones  $\mathcal{Z}2$  de zones  $z_{2,j}$  sans mouvement du robot de durée nominale  $\Delta Tnom_j$  connue nous disposons d'un modèle prédictif permettant d'estimer la durée de la mission  $M$  (Équation 2.8) :

$$Durée(M) = \sum_{i=1}^n \frac{\Delta l_i}{v_{rob_i}} + \sum_{j=1}^m \Delta Tnom_j \quad (2.8)$$

où  $\sum_{i=1}^n \Delta l_i = L$  et  $n = card(\mathcal{Z}1)$  et  $m = card(\mathcal{Z}2)$

En ce qui concerne l'axe Énergie si l'on dispose d'une caractérisation de l'énergie consommée par chaque ressource matérielle, chaque ressource logicielle, ainsi que celle nécessaire au robot pour se déplacer à une vitesse constante, alors il est possible de calculer la puissance  $P(t)$  utilisée à un instant  $t$  par le robot. Dès lors pour une alternative d'implémentation AI donnée sélectionnant les ressources matérielles et logicielles  $P(t)$  est exprimé par la relation suivante (Équation 2.9) :

$$P(AI, v_{rob})(t) = P(Ressources\ matérielles(AI(t))) + P(Ressources\ logicielles(AI(t))) + P(Déplacement(v_{rob}(t))) \quad (2.9)$$

Dès lors si l'on connaît la durée d'application  $\Delta T_i$  de chaque AI tout au long d'un scénario de mission alors il est possible de construire le modèle prédictif de consommation énergétique (Équation 2.10)

$$Énergie(M) = \sum_{i=1}^p P_i(AI_i, v_{rob_i}) \cdot \Delta T_i \quad (2.10)$$

$$\text{avec } p = \text{card}(\mathcal{L}1 \cup \mathcal{L}2)$$

Une explication complète et illustrée de la construction d'un modèle énergétique peut être trouvée dans [113] et [112].

Ce travail préalable d'identification des inducteurs de performance et des modèles prédictifs d'évolution de la performance étant achevé, il est possible de construire le *Scénario de Mission Détaillé (SMD)*.

### 2.3.3 P1 : Phase préliminaire

La phase P1 élabore le scénario de la mission SMD en projetant spatialement l'ensemble des contraintes identifiées (environnementales, expérimentales, etc...) à partir de la description initiale de la mission.

#### 2.3.3.1 Représentation du scénario de la mission

Comme nous l'avons dit plus haut nous disposons initialement :

- Du scénario de la mission décrit comme une succession d'objectifs mettant chacun en œuvre un ensemble de tâches robotiques.
- D'une projection de la mission sur une carte l'environnement.
- Du chemin théorique devant être suivi par le robot.

Cette étape consiste simplement à représenter le long d'un axe correspondant soit au temps écoulé durant la mission mais dont l'unité est pour l'instant arbitraire, où soit à la distance parcourue par le robot, la succession des objectifs et donc des tâches à exécuter (Figure 2.6).

Dans cette figure on constate que l'objectif  $O_{III}$ , pour lequel le robot ne bouge pas, se traduit par un intervalle  $[t_2, t_3]$  selon l'axe du temps et par un unique point  $d_2$  selon l'axe associé à la distance parcourue.

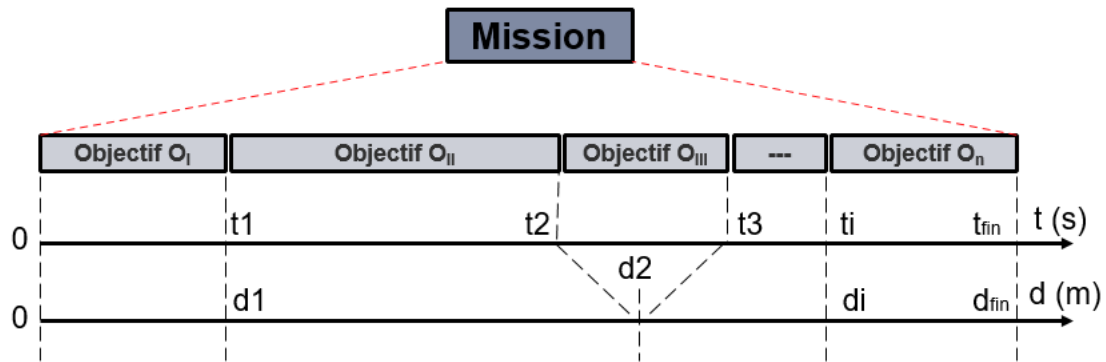


FIGURE 2.6 – Représentation d'une mission

La prochaine étape va consister à projeter les contraintes devant être prises en compte localement sur cette description. Commençons donc par les définir.

### 2.3.3.2 Les contraintes

Nous décomposons les contraintes locales selon 4 types (Figure 2.7) pouvant se superposer en fonction du point de vue retenu :

- Contrainte **environnementale** : Des informations importantes peuvent être extraites de l'environnement comme la position des zones vitrées, sombres, etc... qui peuvent influencer le recrutement de certains capteurs, par exemple, pour des raisons de sécurité.
- Contrainte **méthodologique** : L'analyse du cahier des charges peut induire des contraintes spatiales sur les zones de la carte où une méthode de localisation peut être employée avec succès, ou sur la présence ou l'absence d'humains. Ces contraintes induisent une limitation différenciée des vitesses de déplacement.
- Contrainte **expérimentale** : Évidemment le robot ne peut dépasser une vitesse maximale, et l'utilisation de certains capteurs, sensibles aux vibrations, peut induire une limitation de vitesse lorsqu'ils sont employés.
- Contrainte de **performance** : Les axes de performance Localisation et Sécurité vont induire localement des contraintes sur la vitesse de déplacement ou les méthodes à même de satisfaire les objectifs cœur de mission imposés.

Ces contraintes auront d'une part pour effet, en fonction des inducteurs impactés, de réduire le nombre de ressources matérielles et/ou logicielles éligibles et donc par là même de diminuer les OIT d'une tâche et par conséquent les AI d'un objectif. D'autre part, elles viendront contraindre les paramètres de certaines ressources comme par exemple la vitesse maximale admissible pour le robot  $v_{robmax}$ .

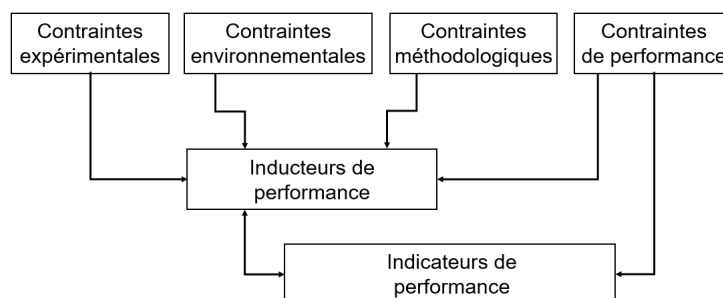


FIGURE 2.7 – Classification des contraintes locales

On peut remarquer que dans la Figure 2.7 la relation bi-directionnelle existant entre les inducteurs et les indicateurs. En effet ces derniers fournissent aussi un cadre d'analyse pour la définition des contraintes imposées aux inducteurs. Par exemple l'indicateur *Innocuité* de l'axe de performance *Sécurité* induit la définition d'une vitesse maximale de déplacement du robot limitant l'énergie cinétique du robot en cas de choc.

### 2.3.3.3 Projection des contraintes sur le scénario : Obtention du SMD

Toutes les contraintes que nous venons d'énoncer ont un champ d'application spatio-temporel plus ou moins limité. Elles sont donc projetées une à une sur la représentation de la mission de façon à identifier, à la fin du processus, des zones à contraintes constantes appelées **Activités** intégrant simultanément l'ensemble des limitations exprimées sur les inducteurs identifiés. Ainsi, pour chacune de ces activités, les ressources restant éligibles sont identifiées ainsi que leurs contraintes d'utilisation ( $v_{robmax}$  par exemple).

Si l'on analyse la Figure 2.8 représentant le mécanisme de projection spatiale ( $[d_{debut}, d_{fin}[$  ou temporelle ( $[t_{debut}, t_{fin}[$ ) on peut observer que la contrainte présentée ne recouvre que partiellement 2 objectifs ( $O_{II}$  et  $O_{III}$ ). Ces derniers se décomposent donc en 4 zones de contraintes constantes correspondant aux activités  $A_{II}$  à  $A_V$ . Par ailleurs on peut aussi constater que l'application de la contrainte ne retient, pour l'activité  $A_{III}$  que 2 des OITs de la tâche  $T_i$  ( $OIT_1$  et  $OIT_3$ ) parmi celles initialement éligibles.

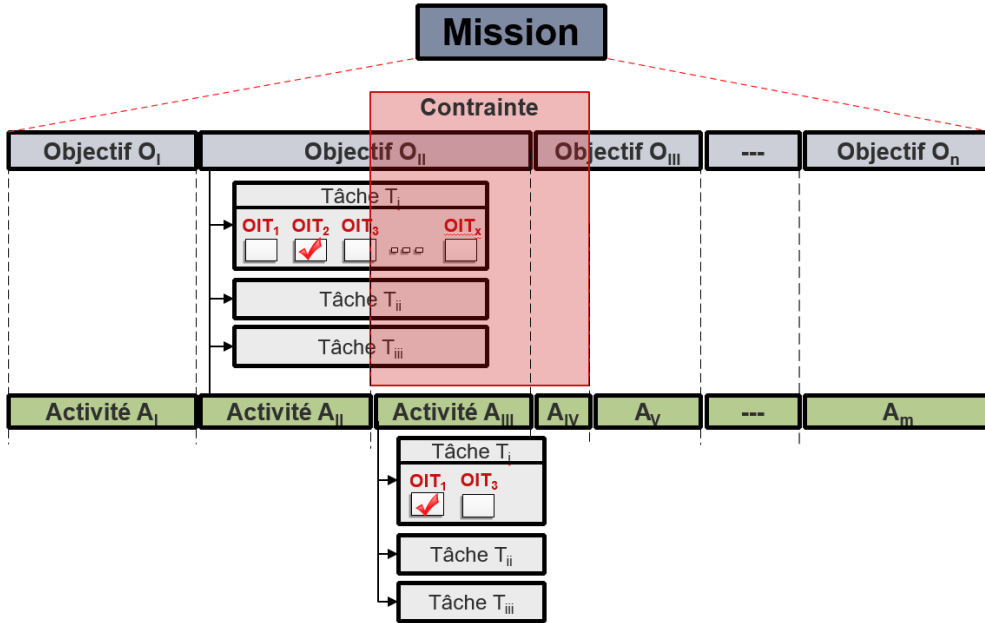


FIGURE 2.8 – Projection des contraintes locales : Activités (d'après [112])

Le SMD correspond donc à une succession de  $n_{Act}$  activités permettant de réaliser la mission définie. Il est alors possible de calculer globalement l'espace d'état  $\mathcal{H}$  au sein duquel il faudra choisir une seule AI par activité pour exécuter la mission. Il s'exprime par l'Équation 2.11.

$$\mathcal{H} = \prod_{i=1}^{n_{Act}} n_{AI}^{A_i} \quad (2.11)$$

où  $n_{AI}^{A_i}$  représente le nombre d'alternatives d'implémentation AI de l'activité  $A_i$

On peut remarquer que la valeur de  $card(\mathcal{H})$  devient très rapidement très grand.

Si les ressources ont été filtrées et paramétrées tout au long de la mission  $M$  pour satisfaire les objectifs de performance cœur de mission, il reste maintenant à choisir localement pour chaque activité d'une mission l'alternative d'implémentation AI qui permettra globalement de satisfaire les objectifs de performance de fin de mission. C'est le rôle de la prochaine phase de la méthodologie PANORAMA.

### 2.3.4 P2 : Estimation hors ligne de la performance

À l'issue de la phase P1, on connaît donc au niveau de chaque activité d'une mission, pour chaque alternative d'implémentation valide, les limites devant être respectées et les valeurs envisageables par les inducteurs de performance qu'il est possible de contrôler.

On s'intéresse maintenant à la performance obtenue globalement en fin de mission. Pour les axes Durée et Énergie, elle s'obtient simplement en sommant les performances de chacune des activités de la mission (Équation 2.12) pour les AIs sélectionnées localement.

$$GPerf_{axe}^{ind}(SMD) = \sum_{i=1}^{n_{Act}} Perf_{axe}^{ind}(A_{Sel_i}) \text{ où} \quad (2.12)$$

*axe* : Représente un axe de performance associé à une performance de fin de mission

*ind* : Représente un indicateur de axe

$A_{Sel_i}$  : Représente l'alternative d'implémentation sélectionnée pour l'activité  $i$  du SMD de la mission

$SMD$  : Représente le Scénario Détaillé choisi pour la Mission (un membre de  $\mathcal{H}$ )

La question est maintenant de déterminer quelle AI doit on choisir localement, au niveau des activités, pour garantir la satisfaction de l'Objectif (contrainte) de performance globale  $Obj_{axe}^{ind}$  imposé. On doit donc vérifier l'équation 2.13.

$$GPerf_{axe}^{ind}(SMD) \leq Obj_{axe}^{ind} \quad (2.13)$$

Le problème à résoudre est donc un problème d'allocation de ressources. Il peut être formulé sous la forme d'un problème de sac-à-dos multi-critères (en Durée et Énergie) [112] qui est connu pour être NP-complexe. Compte tenu de la taille de l'espace de décision ( $\mathcal{H}$ ) il est impossible de trouver la solution optimale. Ainsi dans la Figure 2.9 on trouve sur chaque axe vertical associé à chaque activité du SMD la liste classée (en Durée ou Énergie) des alternatives d'implémentations envisageables (rond). L'algorithme proposé dans [30] [29] permet de trouver une bonne solution au problème de sac-à-dos en un temps polynomial (sélection des ronds jaunes dans la Figure 2.9). C'est lui que nous utilisons en Durée, puis en Énergie pour trouver la **Solution d'Allocation de Ressources** initiale  $SAR_0$  qui sera employée pour la mission.

Pour y parvenir, nous allons adopter des stratégies permettant de guider localement et globalement la sélection des alternatives d'implémentation :

- **Stratégie (Durée)** : *Aller le plus vite possible de façon à chercher à minimiser la Durée de la Mission.*
- **Stratégie (Énergie)** : *Consommer **localement** le maximum d'énergie **TOUT** en vérifiant **globalement** que la contrainte énergétique **reste satisfaite**.*

La stratégie relative à la Durée tombe sous le bon sens en rappelant que les contraintes sécuritaires sont nécessairement satisfaites lorsque l'on aborde la phase P2. En revanche celle relative à l'Énergie est de prime abord contre intuitive puisque les travaux s'intéressant à cet axe de performance cherchent toujours à la minimiser. Cette stratégie a été adoptée car elle va conduire

à sélectionner localement les AI les plus énergivores supposées être les plus performantes, entre autre au niveau de l'axe Localisation qui n'est pas pris en compte explicitement dans [112]. Il ne faut pas oublier que, pour la mission considérée dans sa totalité, on devra s'assurer que la contrainte énergétique reste satisfaite.

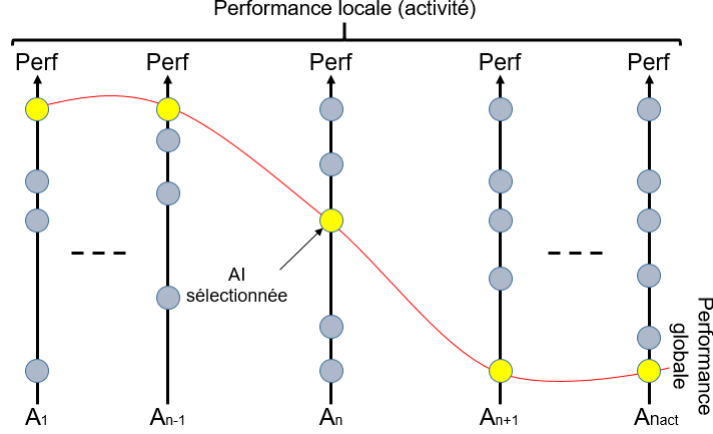


FIGURE 2.9 – Principe d'allocation de ressources

Il est alors intéressant d'introduire la notion de **Marge de Performance**  $M_{axe}^{ind}(t_{obs})$  pour les performances de fin de mission. Evidemment cette marge doit rester positive tout au long de la mission pour que cette dernière soit exécutée avec succès. Soit  $Obj_{axe}^{ind}$  l'objectif de performance à satisfaire et  $\widehat{Perf}_{axe}^{ind}(t_{obs})$  la performance finale estimée à partir de l'instant  $t_{obs}$  en utilisant les modèles prédictifs qui ont été établis, alors on a la condition de l'Équation 2.14 :

$$M_{axe}^{ind}(t_{obs}) = Obj_{axe}^{ind} - \widehat{Perf}_{axe}^{ind}(t_{obs}) \geq 0 \quad \forall axe, \forall ind, \forall t_{obs} \quad (2.14)$$

On peut remarquer que :

- Les choix stratégiques adoptés permettent de maximiser la marge de temps et de minimiser la marge énergétique.
- Les marges constituent des "réservoirs" de performances qui vont permettre d'absorber, au moins momentanément, certains événements imprévus (évitement d'obstacle par exemple) et les imprécisions des modèles prédictifs de performance.

### 2.3.5 P3 : Gestion en ligne de la performance

Durant l'exécution réelle de la mission conformément à la  $SAR_i$  courante, il est possible de relever en permanence à des instants de mesure donnés  $t_{mesure}$  la performance réelle depuis le début de la mission (Durée et Énergie consommées). À partir de cet instant, connaissant  $SAR_i$  et les modèles prédictifs d'évolution de la performance, la performance finale peut être à nouveau estimée et l'évolution des marges de performance peut être suivie.

Si une marge de performance devient négative ou si l'une des alternatives d'implémentations prévues dans  $SAR_i$  devient irréalisable en raison d'une défaillance matérielle ou logicielle, il suffit d'employer la même démarche que celle mise en œuvre dans P2 pour la gestion hors-ligne, mais à partir de l'instant (de l'endroit) où  $SAR_i$  n'est plus applicable, sur la partie de la mission qui n'a pas encore été exécutée. S'il existe une solution permettant de satisfaire tous les axes de performance à l'aide des ressources encore opérationnelles alors on obtient une nouvelle  $SAR_{i+1}$  qui sera alors déployée jusqu'à la prochaine défaillance matérielle/logicielle ou marge de performance négative.

### 2.3.6 Apports et limitations

L'approche méthodologique PANORAMA que nous venons de détailler a été mise en œuvre expérimentalement avec succès [113] [114] [115] [112] sur une mission de patrolling de 200 mètres de long durant 10 minutes. Elle permet de doter le robot d'une autonomie comportementale 'éclairée', nous semble-t-il, présentant un certain nombre d'apports vis-à-vis des travaux existants :

- Elle assure la prise en compte simultanée de plusieurs axes de performance et particulièrement de la dimension énergétique qui est rarement traitée avec autant de précision.
- Elle est assez robuste aux erreurs de connaissance (trajectoire suivie différente de celle envisagée, modèle énergétique imparfait) grâce aux marges de performance.
- Elle permet de doter le robot d'un bon niveau de tolérance aux fautes en proposant des solutions alternatives de reconfiguration.
- L'efficacité de l'algorithme d'allocation mis en œuvre sur un problème NP-hard permet de trouver des solutions en ligne.

Malgré ces apports, la méthodologie PANORAMA présente aussi un ensemble de limitations :

- Le mécanisme d'identification des activités du SMD par projection des contraintes est complexe à mettre en œuvre.
- En cas de reconfiguration en ligne, la multiplication des activités et des alternatives d'implémentations peut conduire à un espace d'états dont la dimension ne permettra plus une mise en œuvre en ligne, durant l'exécution de la mission, de l'algorithme d'allocation des ressources.
- La logique de minimisation de la marge énergétique adoptée va nécessairement multiplier les reconfigurations en raison des inévitables erreurs de modèle.
- L'axe de performance Localisation est pris en compte de façon qualitative sans assurer à l'utilisateur qu'une marge de localisation fixée au préalable reste bien satisfaite.

Le cadre général dans lequel et à partir duquel les travaux présentés dans ce manuscrit s'inscrivent étant maintenant présentés nous allons préciser la problématique scientifique qui va être abordée.

## 2.4 Problématique scientifique de la thèse

Comme nous venons de l'indiquer, la proposition méthodologique actuelle ne permet que de prendre en compte partiellement le problème de garantie de localisation puisqu'elle suppose d'une part que les algorithmes qui peuvent être utilisés au sein des activités sont immédiatement opérationnels, et que, d'autre part, ces algorithmes permettent d'obtenir une localisation du robot adaptée tant à la tâche devant être exécutée, qu'au déroulement attendu de la mission. Ces suppositions relèvent plutôt du postulat et de la qualité de l'expertise de la personne qui a décrit la mission à réaliser. L'objectif du travail présenté dans ce document sera de tenter d'apporter, tout au long de la mission, un choix plus objectif des méthodes de localisation pouvant et devant être mises en œuvre et cela sans pour autant déroger à la satisfaction des autres contraintes de performance.

Qui plus est, cette objectivation du choix des méthodes de localisation va nous permettre de basculer, si l'utilisateur le souhaite, vers une nouvelle stratégie d'affectation des ressources en permettant de moduler, voire même de maximiser la marge énergétique au lieu, comme cela

est le cas actuellement, de la minimiser. En effet, il faut préciser pourquoi le choix de la maximisation énergétique, contre intuitif, a été retenu dans les travaux précédents pour garantir une consommation d'énergie inférieure à un seuil fixé. Cette stratégie, totalement arbitraire, postule que plus l'énergie consommée est importante, plus les méthodes de localisation employées sont performantes. Cela favorise donc celles nécessitant un temps de calcul important et/ou faisant appel à de nombreux capteurs. Si la maximisation de la marge énergétique avait été adoptée, l'algorithme aurait nécessairement sélectionné comme méthode de localisation, pour toutes les activités, l'odométrie qui est toujours accessible sans consommation d'énergie supplémentaire. Or, il est notoire que cette méthode de localisation dérive avec la distance parcourue. Le robot se serait ainsi rapidement perdu en se croyant à une position très éloignée de la réalité. Cela aurait donc conduit le robot à engager des actions hors des zones au sein desquelles elles étaient planifiées. De façon à pouvoir s'engager dans une logique permettant d'augmenter la marge énergétique il faut donc aller plus loin dans l'identification des méthodes de localisation pouvant être adoptées lors de l'exécution d'une tâche robotique en étant capable de ne sélectionner que celles qui seront à même de garantir la précision de localisation imposée par l'utilisateur.

Compte tenu des objectifs qui viennent d'être énoncés les principales problématiques scientifiques adressées dans le cadre de cette thèse sont donc les suivantes :

- Comment proposer, pour les méthodes de localisation choisies, un modèle prédictif de l'évolution de l'incertitude de localisation ?
- Comment intégrer explicitement, dans la méthodologie PANORAMA, la prise en compte d'une garantie de localisation ?
- Comment basculer, dans la méthodologie PANORAMA, d'une logique de minimisation de marge énergétique, vers une logique de modulation, voire de maximisation de cette dernière ?

## 2.5 Conclusion

Cette première partie a tout d'abord permis de présenter les concepts d'autonomie et de performance avant de s'attarder sur la problématique de garantie de performance. Puis, la méthodologie PANORAMA, développée au sein de l'équipe EXPLORE, a été détaillée. Elle permet d'assurer, au niveau mission, la satisfaction d'objectifs de performance, même en présence de fautes matérielles et/ou logicielles, lors de l'exécution d'une mission en gérant à la volée les ressources utilisées et leur paramétrage. Elle constitue le socle sur lequel viendra s'articuler le travail présenté dans le cadre de ce manuscrit.

Le prochain chapitre va maintenant se focaliser sur les fondements problématiques de localisation en robotique mobile qui constituent le second corpus scientifique dont relève le travail présenté dans ce manuscrit.





# Chapitre 3

## LA LOCALISATION EN ROBOTIQUE MOBILE : ÉLÉMENTS, PROBLÉMATIQUES ET APPROCHES

Ce chapitre vise à donner une vision générale du problème de localisation en robotique mobile. Nous commencerons par rappeler quelques éléments fondamentaux en présentant comment les architectures matérielle et logicielle se conjuguent pour permettre au robot de se déplacer. Nous porterons une attention toute particulière aux capteurs et à leurs mesures qui jouent un rôle central dans notre étude. Dans un second temps nous déclinerons les différentes problématiques de localisation étudiées en robotique mobile. On se localise toujours dans ou par rapport à un environnement, c'est pourquoi nous en balaiersons ensuite les principales approches de représentation. La partie suivante expose rapidement les principes des méthodes de localisation développées en robotique mobile. Pour terminer nous identifierons l'origine des incertitudes venant entacher le processus de localisation avant d'analyser les performances affichées par un ensemble de travaux représentatifs du domaine.

### 3.1 Éléments de robotique mobile

Cette partie présente les principaux éléments, tant matériels que logiciels, constitutifs d'un robot mobile ainsi que les référentiels et les notations qui seront utilisés pour le localiser.

#### 3.1.1 Quelques définitions

Un **robot mobile terrestre**, supposé rigide, est "*une machine autonome équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a*" (d'après [75]).

Comme le montre la Figure 3.1, par l'intermédiaire de ses capteurs, les capacités de perception du robot lui permettent d'obtenir des informations sur son état interne ainsi que sur son environnement pouvant lui même être instrumenté (amers artificiels). Sa base mobile le dote de la capacité de se mouvoir en contrôlant ses actionneurs en fonction des objectifs qui lui ont été assignés. Enfin ses capacités décisionnelles, implantées logiquement au sein de contrôleurs, constituent le chaînon central du cycle Perception / Décision / Action.

Pour que le robot puisse se déplacer de façon adaptée à ses objectifs dans son environnement, il est indispensable de définir un ensemble de **repères / référentiels** qui seront utilisés pour le localiser dans l'espace. Dans le cadre de ce document, nous nous limiterons, à un espace euclidien à 2 dimensions doté de repères cartésiens orthonormés, c'est à dire à une localisation

dans le plan, Comme il sera mentionné plus loin, certaines méthodes de localisation vont nécessiter un référentiel 3D, mais l'objectif reste une localisation 2D.

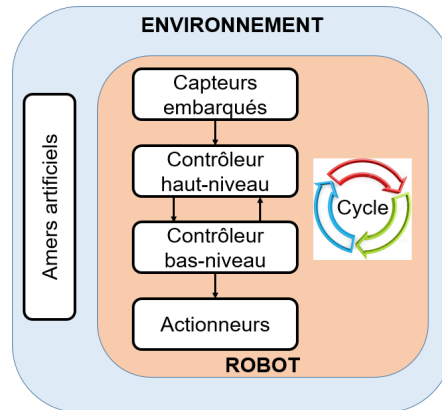


FIGURE 3.1 – Le robot dans son environnement

On définit alors les référentiels cartésiens  $\mathcal{R}_k$  d'origine  $O_k$  et de base orthonormée  $\mathcal{B}_k = (\vec{i}_k, \vec{j}_k)$  suivants (Figure 3.2) :

- **Repère Monde** : C'est un repère fixe  $\mathcal{R}_M$ , d'origine  $O_M$  et de base  $\mathcal{B}_M$  associé à la carte de l'environnement dans lequel évolue le robot.
- **Repère Robot** : C'est un repère mobile  $\mathcal{R}_R$ , d'origine  $O_R$  et de base  $\mathcal{B}_R$  associé au robot. Le plus souvent son origine est placée au centre de gravité, ou au centre de rotation du robot.
- **Repère Capteur** : C'est un repère  $\mathcal{R}_c$ , d'origine  $O_c$  et de base  $\mathcal{B}_c$ . Nous distinguerons 2 cas de figures :
  - C'est un référentiel associé à un capteur du robot. Il est donc fixe dans  $\mathcal{R}_R$  et mobile dans  $\mathcal{R}_M$ .
  - C'est un référentiel associé à un amer de l'environnement. Il est donc fixe dans  $\mathcal{R}_M$  et si c'est un amer artificiel il peut être géo-référencé dans  $\mathcal{R}_M$ .

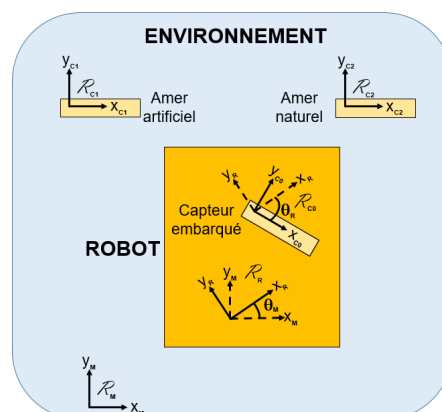


FIGURE 3.2 – Les référentiels

En fonction de ce qui est recherché, il est possible de localiser le robot relativement à son environnement, ou globalement, en déterminant sa localisation absolue dans  $\mathcal{R}_M$ , au sein d'une carte, dans le repère Monde.

Il est donc nécessaire, d'effectuer des changements de repère afin de pouvoir, en fonction des besoins, localiser le robot dans un référentiel cible. Ainsi, dans le cas général, 2 repères peuvent être définis l'un par rapport à l'autre par le biais d'une transformation rigide linéaire homogène combinant une rotation et une translation selon l'Équation 3.1. Soit un point P de coordonnées cartésiennes  $P|_{R_1} = \begin{bmatrix} x_{R_1} \\ y_{R_1} \\ 1 \end{bmatrix}^T$  dans un repère  $\mathcal{R}_1$  et  $P|_{R_2} = \begin{bmatrix} x_{R_2} \\ y_{R_2} \\ 1 \end{bmatrix}^T$  dans un repère  $\mathcal{R}_2$  alors on a en passant en coordonnées homogènes :

$$P|_{R_2} = \begin{bmatrix} x_{R_2} \\ y_{R_2} \\ 1 \end{bmatrix} = \begin{bmatrix} R(\alpha) & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_{R_1} \\ y_{R_1} \\ 1 \end{bmatrix} = MT_{R_2/R_1} \bullet \begin{bmatrix} x_{R_1} \\ y_{R_1} \\ 1 \end{bmatrix} \quad (3.1)$$

$MT_{R_2/R_1}$  : Représente la matrice de transformation homogène de passage de  $\mathcal{R}_1$  à  $\mathcal{R}_2$

$R(\alpha)$  : Représente la matrice de rotation de  $\mathcal{R}_1$  par rapport à  $\mathcal{R}_2$

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (3.2)$$

$T$  : Représente une translation qui exprime les coordonnées de  $O_{R_1}$  dans  $\mathcal{R}_2$

$0^T = [0 \ 0]$  représente un vecteur nul

**La localisation** du robot (Équation 3.3) à l'instant  $t$  correspond alors aux coordonnées cartésiennes de l'origine  $O_R$  du repère  $\mathcal{R}_R$  exprimées dans le référentiel Monde  $\mathcal{R}_M$ .

$$O_R(t)|_M = \begin{bmatrix} x_R(t)|_M, y_R(t)|_M \end{bmatrix}^T \quad (3.3)$$

Si l'on adjoint l'angle de rotation  $\theta_R$  ou orientation du repère du robot  $\mathcal{R}_R$  par rapport au repère Monde  $\mathcal{R}_M$  on parle alors de **pose** PO du robot (Équation 3.4) à l'instant  $t$ .

$$PO_R(t)|_M = \begin{bmatrix} x_R(t)|_M, y_R(t)|_M, \theta_R(t)|_M \end{bmatrix}^T \quad (3.4)$$

Dans la suite du document nous utiliserons souvent de façon indifférenciée les termes position, localisation et pose pour désigner le lieu où se trouve le robot. Les définitions de base étant rappelées nous allons maintenant préciser la composition et le rôle des différents composants d'un robot mobile en adoptant une vue matérielle et logicielle.

## 3.1.2 Vue matérielle

### 3.1.2.1 Les capteurs

Les capteurs constituent la première composante de la vue matérielle d'un robot. Un capteur est un système de mesure qui permet de transformer un stimulus physique, en un signal, le plus souvent électrique, sous la forme d'un courant ou d'une tension. La mesure obtenue est toujours bruitée et peut être être corrompue par des erreurs. Elle n'est donc jamais parfaite et est toujours entachée d'incertitudes.

En robotique, en fonction des informations qu'ils permettent de recueillir, on distingue 3 grandes classes de capteurs, les capteurs proprioceptifs, intéroceptifs et extéroceptifs (Figure 3.3).

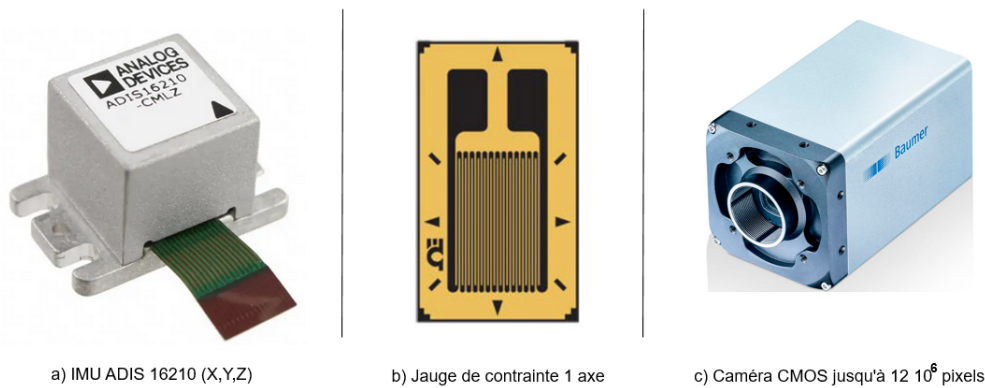


FIGURE 3.3 – Exemples de capteurs :  
a) Proprioceptif - b) Intéroceptif - c) Extéroceptif

**Les capteurs proprioceptifs** permettent d’obtenir des informations liées à l’état interne du robot et plus particulièrement l’accès à ses paramètres cinématiques (position, vitesse, accélération). On distingue les *capteurs de déplacement* tels que les odomètres ou les accéléromètres qui, à partir de la mesure de vitesse (des roues) ou d’accélération, autorisent, par intégration, la reconstruction de la trajectoire curviligne du robot. Il existe aussi des *capteurs d’attitude* tels que les gyroscopes, gyromètres ou inclinomètres qui servent à retrouver les angles de cap, roulis et tangage du robot. Les informations manipulées étant fortement bruitées elles conduisent inévitablement, dans le temps, au non bornage de l’erreur associée au calcul incrémental du déplacement du robot. En effet, en l’absence d’informations le relocalisant relativement à son environnement, le robot ne compte que sur sa perception interne pour évaluer son déplacement et son attitude, on parle alors de navigation à l’estime. En revanche les informations obtenues présentent deux avantages notables : elles sont assez peu influencées par les conditions expérimentales (nature de l’environnement, éclairage, etc.), et restent assez simples à obtenir.

**Les capteurs intéroceptifs** sont rarement distingués des capteurs proprioceptifs dans la littérature puisqu’ils concourent aussi à renseigner sur l’état interne du robot. Ils ciblent cependant des informations de nature différente souvent très utiles pour évaluer globalement la sûreté du système robotique. On peut citer par exemple les capteurs de température interne qui peuvent détecter une évolution anormale de cette dernière, les détecteurs d’arrivée d’eau indispensable pour les robots sous-marins, les jauges de contraintes qui judicieusement placées alertent sur un niveau anormal d’effort au sein de la structure, enfin évidemment les capteurs de niveau de charge des batteries (wattmètre embarqué ou autres) qui sont indispensables pour gérer la dimension énergétique d’une mission autonome.

**Les capteurs extéroceptifs** sont des capteurs qui mettent en œuvre des principes physiques nécessitant une interaction avec l’environnement. On parle souvent de *perception*. Ils permettent de replacer le robot relativement à l’environnement. Pour déterminer la distance à un obstacle les *capteurs télémétriques* calculent le temps de vol (temps aller/retour) d’ondes qu’ils émettent dans leur environnement selon une direction. On distingue principalement les sonars (ou capteur à ultrasons) qui utilisent des ondes ultra-sonores, les radars qui font appel à des ondes électromagnétiques et les télémètres laser qui emploie un faisceau laser. Ils possèdent cependant des caractéristiques très différentes tant en terme de distance mesurable, de rapidité, de précision, de sensibilité aux conditions expérimentales et de coût. Il faut aussi ajouter des *capteurs de contact* (bumpers), qui permettent de détecter la collision, ou la forte proximité (moustaches) du robot avec un obstacle et permettent donc d’engager une action d’urgence. On peut aussi faire appel à des *systèmes de vision* pour obtenir des informations sur l’environne-

ment. Une camera, le plus souvent CCD (Charged Coupled Device) ou CMOS (Complementary Metal-Oxide-Semiconductor) permet d'obtenir une représentation pixélisée de l'environnement à partir de laquelle il est possible d'extraire une information 2D. L'adjonction d'une seconde caméra permet par analyse stéréoscopique de calculer la distance à un *amer*. Dans l'environnement ces derniers peuvent être *naturels* ou *artificiels*. Les premiers constituent des points caractéristiques (portes, fenêtres, angles, etc.) déjà présents, alors que les seconds sont ajoutés, géo-référencés et contiennent une information codée (QR-Code par exemple). Les systèmes de vision sont en général performants tant en terme de rapidité que de précision. En revanche ils nécessitent la manipulation et le traitement d'un flux de données très important souvent coûteux en temps de calcul.

Il faut aussi ajouter que le robot, au cours de son évolution, peut aussi faire appel à des amers *artificiels actifs* disséminés et géo-référencés dans l'environnement. Ces balises émettent des signaux codés (infrarouge ou radio) qui sont détectés puis analysés par le système de mesure embarqué. Le système GPS (Global Positioning System) utilisé en extérieur relève de ce principe de géo-localisation. Dans ce cas les satellites jouent le rôle de balises émettrices et la position du robot est ensuite obtenue en faisant appel à une méthode de triangulation sophistiquée. La mise en place de balises artificielles au sein de l'environnement s'avère peu flexible et peut être coûteux. En revanche, si la géo-localisation a été effectuée avec soin, cela permettra une localisation précise et robuste du robot en absolu, dans le repère Monde.

### 3.1.2.2 Actionneurs et effecteurs

Les actionneurs permettent à un système robotique de se mettre en mouvement et d'interagir physiquement avec son environnement. En robotique, les actionneurs sont le plus souvent des moteurs ou des servo-moteurs à courant continu ou brushless commandés à l'aide de PWM (Pulse Width Modulation) qui assurent la conversion d'une énergie électrique en énergie mécanique pour commander un déplacement.

Dans ce paragraphe par abus de langage nous présentons plus largement les architectures mécaniques usuelles proposées en robotique mobile terrestre (Figure 3.4).

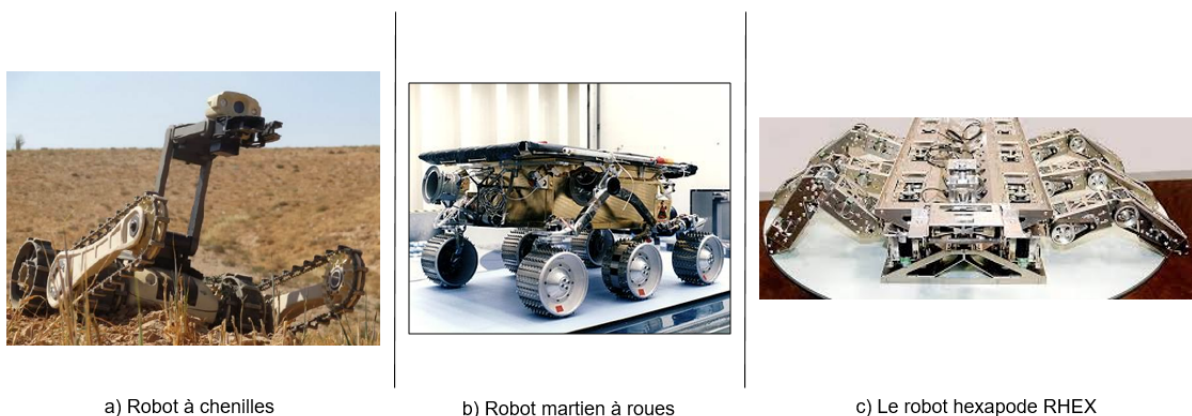


FIGURE 3.4 – Exemples de robots terrestres

Si l'on se focalise sur la base mobile qui permet au robot de se mouvoir, de multiples approches de conception mécanique ont été proposées mettant en œuvre différents types de robots en fonction de la nature de l'environnement (extérieur ou intérieur) et des capacités de franchissement visées. La propriété d'**holonomie** qui caractérise l'aptitude à pouvoir se déplacer instantanément dans toutes les directions est une des caractéristiques notables des

robots mobiles. Pour ce faire il faut que le robot dispose d'au moins autant de degrés de liberté contrôlables que le nombre total de degré de liberté. Dans le plan, un robot doit donc avoir 3 degrés de liberté : 2 translations et une rotation.

En environnement intérieur l'**effecteur** naturellement utilisé est la roue. On distingue alors 3 classes principales de robots à roues [223] :

- Robot **unicycle** : Il est composé de 2 roues contrôlées indépendamment et d'au moins une roue folle. Cette plate-forme différentielle est non-holonome. Ses avantages majeurs sont sa simplicité de conception et de contrôle.
- Robot **Ackerman** : C'est un robot de type voiture ne possédant qu'un système de roues motrices à braquage différentiel à l'avant uniquement. Les conducteurs savent que ce robot n'est pas holonome et que donc toutes les trajectoires ne sont pas envisageables.
- Robot **omnidirectionnel** : La propriété d'holonomie peut être obtenue en faisant par exemple appel à des roues dites "suédoises" disposées au sommet d'un triangle et dont on contrôle indépendamment la vitesse pour obtenir la trajectoire voulue, ou en utilisant directement des roues dont on peut contrôler la vitesse de rotation et la direction.

En terrain extérieur, en raison des glissements, les roues sont souvent abandonnées pour faire appel à des chenilles en s'appuyant le plus souvent sur une conception de type robot unicycle. On peut aussi utiliser comme effecteurs des pattes (multi-podes) ce qui confère au robot des capacités de franchissement intéressantes et minimise son impact au sol.

Pour interagir avec son environnement il faut adjoindre à la base mobile un bras manipulateur [36] et gérer simultanément, si nécessaire, le déplacement du robot ainsi que le positionnement et la force exercée par l'effecteur (pince, etc.) placé à l'extrémité du bras.

### 3.1.2.3 Organisations matérielle et spatiale

Même si les capteurs et les actionneurs sont extrêmement importants pour un robot, un robot ne se réduit pas à ces deux seules classes de composants. En premier lieu il ne faut évidemment pas occulter les processeurs au sein desquels les informations capteurs sont traitées pour alimenter les algorithmes de commande qui fournissent une consigne aux actionneurs. La fréquence de leur horloge (modifiable ou non), la capacité mémoire embarquée, sont autant de paramètres qui influenceront les performances du robot, dont sa consommation énergétique. Pour un système autonome, il faut aussi ajouter les batteries dont le choix influencera, son poids, son volume, son rayon d'action et la durée de rechargement. Enfin, il est aussi possible d'ajouter des organes tels que des commutateurs contrôlables (*switch*) qui peuvent permettre, en ligne, de connecter, ou non, les organes qu'ils alimentent.

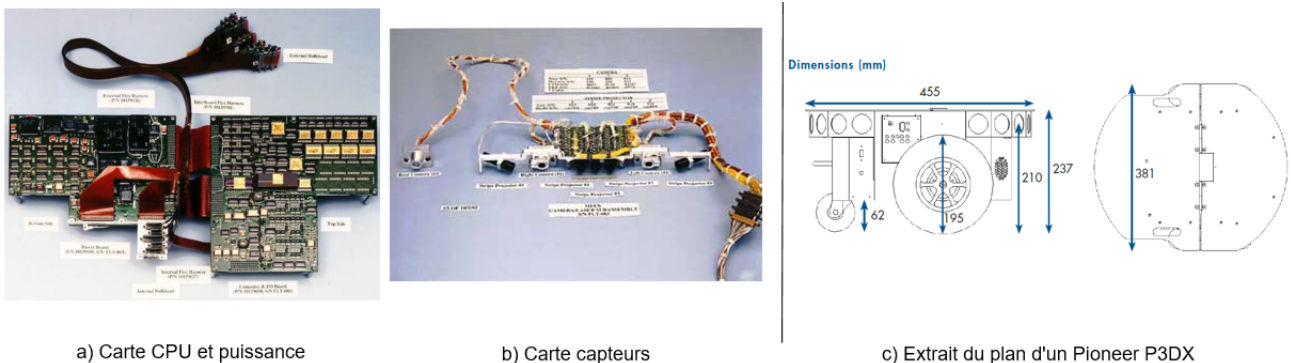


FIGURE 3.5 – Exemples d'organisations matérielles : a) et b) Architecture électronique du robot martien *Pathfinder* - c) Plan d'un robot

En adoptant le point de vue matériel, il nous semble indispensable de retenir deux facettes complémentaires pour décrire physiquement l'organisation interne d'un robot (Figure 3.5). D'une part une dimension purement architecturale mettant en avant la nature des différents éléments constitutifs d'un système robotisé que sont capteurs, actionneurs, processeurs et batteries ainsi que leurs relations au sein d'une représentation abstraite. La connaissance de ces liens, ainsi que de leur dynamicité est essentielle pour maîtriser la dimension énergétique d'une mission autonome. D'autre part une vue purement mécanique et dimensionnelle (métrologique) positionnant spatialement ces éléments dans l'espace lié au référentiel robot. Ces informations, particulièrement au niveau du positionnement des capteurs et des effecteurs au sein du robot, ainsi que de la précision associée, seront particulièrement importantes pour reconstruire la position et la pose du robot.

Après avoir présenté synthétiquement les différents constituants de la vue matérielle d'un robot, nous allons maintenant le décrire au travers de sa vue logicielle.

### 3.1.3 Vue logicielle

L'architecture logicielle d'un robot ou **architecture de contrôle** permet d'organiser logiquement, temporellement les relations devant exister entre un ensemble d'entités logicielles et matérielles pour qu'un robot puisse atteindre les objectifs qui lui ont été fixés.

En fonction de l'organisation choisie, 3 classes architecturales (Figure 3.6) ont été proposées en robotique [183] [243] :

- **Réactive / Comportementale** : Ce type d'architecture s'appuie sur un ensemble de comportements élémentaires, s'exécutant simultanément, soumis aux stimuli de l'environnement, qu'un module d'arbitrage combine de façon plus ou moins complexe [43] [231] pour générer la commande courante la plus adaptée. Cette approche minimise le temps de réaction entre perception et action. Elle ne nécessite pas de plans de mission et ne fait appel à aucune modélisation et mémorisation de l'environnement. Ces choix la rendent inadaptée pour mener à bien des missions complexes nécessitant une stratégie élaborée.
- **Délibérative** : A contrario cette classe architecturale décompose les mécanismes décisionnels au travers de 3 couches aux horizons temporels distincts :
  - La couche **décisionnelle** qui planifie l'enchaînement des tâches robotiques à mettre en œuvre pour répondre aux objectifs spécifiés.
  - La couche **exécutive** qui supervise l'exécution des tâches robotiques courantes.
  - La couche **fonctionnelle** où l'on trouve l'ensemble des modules logiciels mobilisés par la couche exécutive. Cette couche assure l'interface avec le monde physique, à savoir avec les capteurs pour la perception, et les actionneurs pour la commande.

Cependant de part la direction des liens logiques existants entre ces différentes couches, cette architecture présente un manque de réactivité incompatible avec une utilisation en milieu fortement dynamique.

- **Hybride** : En ajoutant des boucles de réaction inter-couches à une structure délibérative on obtient alors une organisation logicielle associant simultanément réactivité et capacité décisionnelle. Cette classe architecturale est actuellement largement utilisée en robotique [226] [108] [72].

La mise en œuvre d'une architecture de contrôle robotique nécessite de suivre un certain nombre de règles de développement classiques en informatique afin de satisfaire un ensemble de propriétés non fonctionnelles facilitant sa mise en œuvre, sa maintenance et la réutilisation du code développé. Parmi les plus classiques on peut citer la modularité, la flexibilité, la réutilisabilité ou la fiabilité, par exemple. Pour faciliter leur développement il est possible de faire appel



à des **middlewares**. Ces entités logicielles servent d'interface entre l'application et le système d'exploitation sur lequel cette dernière est exécutée, et facilite la mise en œuvre des communications entre les différents composants de l'architecture. Le plus utilisé en robotique est ROS (*Robot Operating System*) [189] [134] pour lequel chacun des nœuds correspond classiquement à une boucle temporelle (tâche) donnée. Son atout majeur est de disposer d'un ensemble très important de bibliothèques de drivers de capteurs et/ou d'actionneurs ou de tâches robotiques. Par ailleurs il peut facilement être interfacé avec d'autres langages. Cependant il ne permet pas une maîtrise fine des pipelines logiques et n'apporte pas de garantie temps réel qui peut parfois être exigée pour des applications robotiques. Cette dimension est adressée par certains middlewares comme OROCOS [210] pouvant être associés à un langage de modélisation comme MAUVE de l'ONERA [91], qui portent une attention particulière à la maîtrise de l'exécution temporelle des tâches logicielles. CONTRACT [182], développé au LIRMM, permet en plus une reconfiguration dynamique, "à la volée", du pipeline logique des composants devant être exécutés au sein d'une boucle temporelle.

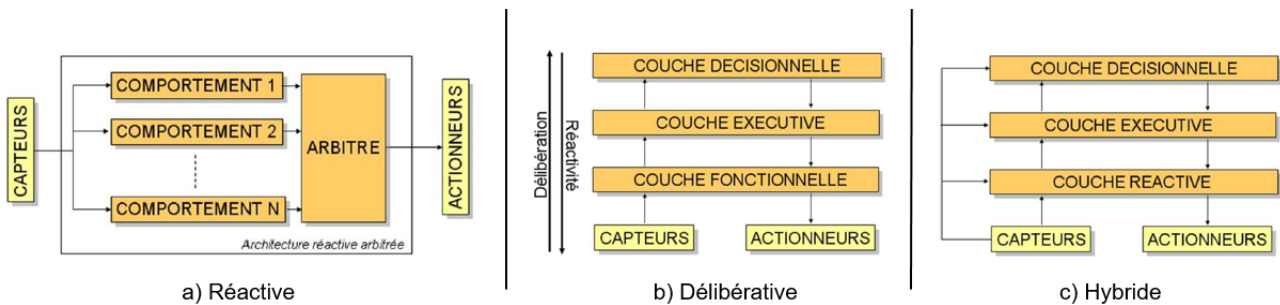


FIGURE 3.6 – Principales classes architecturales robotiques

L'architecture de contrôle assure nécessairement l'interface entre le monde réel nécessairement continu et le monde numérique au sein duquel les informations de perception sont traitées de façon à pouvoir générer des commandes d'actionneurs permettant au robot d'atteindre les objectifs qui lui ont été imposés en adoptant une stratégie adaptée. Ce changement de monde conduit inévitablement à un échantillonnage des données, qui, ajouté aux inévitables délais de traitement de ces informations, induit une discrétisation des commandes calculées. Qui plus est, pendant chaque cycle de contrôle-commande, le robot reste le plus souvent en mouvement. Cela induit par la même que certaines des informations manipulées, et particulièrement celles liées à la connaissance de l'environnement, reflètent une situation passée et sont donc bruitées et entachées d'incertitudes et de retards. L'impact de ces dernières se propagera tout au long de la chaîne de génération des commandes.

L'architecture de contrôle, actuellement le plus souvent modulaire et à composants, doit être à même de mettre en œuvre la chaîne de *Navigation - Guidage - Contrôle* que l'on trouve classiquement au sein de nombreux robots mobiles. Cette stratégie permet au robot de se déplacer de sa position courante vers un objectif visé en faisant appel à une trajectoire précalculée. Elle est décomposée (Figure 3.7) dans le temps de :

- La **Navigation** : D'après [105] la Navigation peut être définie comme étant "*le processus permettant de décider et contrôler la direction de déplacement, à partir de la connaissance de la localisation du robot au sein d'une carte*". Cela nécessite donc de savoir 1) Où se trouve le robot, 2) Où se trouvent, par rapport au robot, les points de passage qu'il doit rallier, 3) Comment il peut les rallier depuis sa position courante [145]. Classiquement, à partir de la position initiale du robot, de la connaissance de la carte et des points de passage, le module Navigation planifie la trajectoire que devra suivre le robot.

- Le **Guidage** : Le Guidage, à partir de la connaissance de la position courante du robot par rapport à la trajectoire, et du prochain point de passage à rallier, détermine les valeurs des vitesses linéaires et angulaires à imposer pour suivre la trajectoire définie. On parle alors de *suivi de chemin*.

Pour ce faire on fait appel au *modèle cinématique* du robot, qui permet de décrire son mouvement indépendamment des causes le produisant. Dans ce cas on s'intéresse donc aux grandeurs cinématiques que sont la position, la vitesse et l'accélération. Il est aussi possible d'employer le *modèle dynamique* qui s'intéresse en plus de la cinématique aux forces qui ont engendré le mouvement. Les grandeurs dynamiques considérées sont alors les forces, accélérations, et moments.

Le module de Guidage intègre aussi, la plupart du temps, un mécanisme d'évitement d'obstacle permettant au robot de modifier localement sa trajectoire de façon à éviter un obstacle dynamique qui n'aurait pas été pris en compte par le module de Navigation.

- Le **Contrôle** : Enfin le Contrôle asservit les commandes actionneurs (moteurs) aux vitesses définies par le module de Guidage.

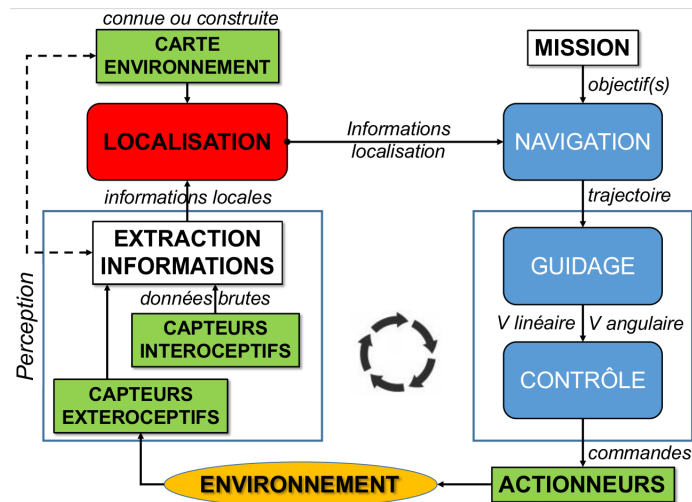


FIGURE 3.7 – Localisation et séquence Navigation-Guidage-Contrôle (inspiré de [168] [204])

Ce paragraphe a rapidement présenté les différents principes de mise en œuvre d'une architecture de contrôle robotique. La définition du rôle des modules de Navigation et de Guidage a mis en évidence que la localisation du robot était un pré-requis au déplacement du robot. Nous allons maintenant focaliser notre discours sur cette tâche centrale de la robotique.

## 3.2 Problématiques de localisation

La Localisation est le "*processus permettant de reconstruire ou d'identifier la pose d'un robot mobile au sein d'une carte*" [105]. La Figure 3.7 synthétise son positionnement et ses interactions avec les différents modules de Navigation, Guidage et Contrôle. Elle montre clairement sa position centrale dans le processus de déplacement d'un robot. On y voit apparaître aussi un élément important sur lequel nous reviendrons plus en détail dans le prochain paragraphe, l'utilisation d'un modèle de représentation de l'environnement. En effet on se localise toujours par rapport à quelque chose et donc il faudra distinguer la *localisation locale* ou *relative* du robot vis-à-vis de son environnement proche, de la *localisation globale* ou *absolue* dans le référentiel monde.

Classiquement, lorsque l'on dispose d'une connaissance à priori de l'environnement par le biais d'une carte, on distingue deux problématiques de localisation, le suivi de posture et la localisation globale.

Le **suivi de posture** (Pose Tracking), on parle aussi de **localisation locale**, correspond à une situation couramment rencontrée en robotique où la location initiale du robot est connue avec une incertitude faible. Tout au long de sa mission, le robot va mettre à jour en permanence sa position à partir de sa dernière actualisation, en estimant les déplacements qu'il réalise (Figure 3.8 a ). En ne se basant que sur ses capteurs proprioceptifs, c'est à dire sans aucune information sur sa position par rapport à l'environnement, l'estimation de la localisation du robot va se dégrader rapidement. Pour pallier cette difficulté, et endiguer cette inexorable dérive, il est alors possible d'adjoindre, lorsque cela est possible, un recalage de la pose du robot à partir d'informations extéroceptives obtenues à partir d'amers naturels ou artificiels de l'environnement géo-référencés, permettant de relocaliser globalement le robot dans la carte. Enfin, on peut aussi faire appel à des approches probabilistes qui, en s'appuyant sur la connaissance de la carte, les informations extéroceptives, la connaissance de la commande du robot et les incertitudes associées, va permettre de suivre la posture du robot tout au long de son évolution, avec une certaine précision.

La **localisation globale** considère que la position initiale du robot est inconnue (Figure 3.8 b). C'est une situation qui peut se rencontrer en robotique autonome où le robot ne peut avoir recours à une aide extérieure pour se localiser. Ce problème est bien plus difficile que le précédent car, sans connaissance sur sa position initiale, le robot doit se repérer, évidemment en s'aidant des informations extéroceptives qu'il peut collecter, mais au sein d'un environnement qui peut présenter certaines similitudes. La localisation globale peut donc être utilisée pour connaître la position initiale d'un robot et ensuite effectuer un suivi de posture.

A ces deux principales classes de problématiques de localisation s'ajoute celle du **kidnapping** qui relève en fait des deux précédentes. Dans ce cas on considère que le robot préalablement localisé est soudainement déplacé (téléporté) quelque part dans l'environnement (Figure 3.8 c)). Il doit donc être capable de détecter qu'il a subi un déplacement inapproprié et qu'il a été kidnappé. On se retrouve alors face à un problème classique de localisation globale.

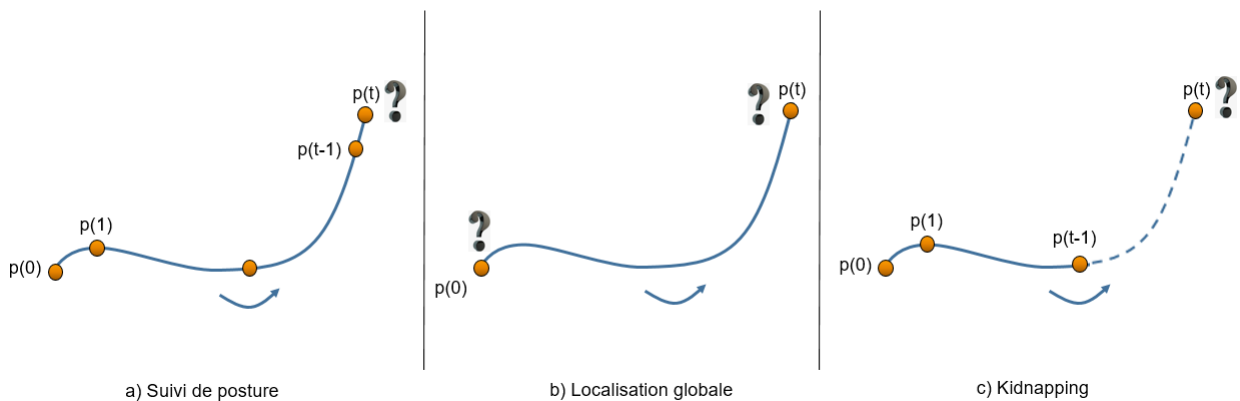


FIGURE 3.8 – Les problématiques de localisation (inspiré de [168])

$p(i)$  : Représente la pose du robot à  $i$   
 $p(0)$  : Représente donc la pose initiale du robot  
 $p(t)$  : Représente donc la pose courante du robot

La localisation s'effectuant nécessairement au sein d'un environnement, nous allons maintenant nous attarder sur les différents modèles pouvant être employés pour le modéliser.

## 3.3 Représentations de l'environnement

Qui dit se localiser, dit nécessairement disposer d'une carte de l'environnement. Dans un premier temps, et avant d'aborder les modèles pouvant être utilisés, il est important de distinguer et de classer ces derniers.

### 3.3.1 Une typologie des environnements

De premier abord il est évident qu'il faut les différencier en fonction de leur variabilité. Les environnements **statiques** possèdent des caractéristiques tant dimensionnelles, qu'en terme de caractéristiques (éclairage par exemple) qui restent invariantes au cours du temps. Cela exclut aussi bien évidemment la présence au sein de la carte d'objets inconnus et mobiles. On voit bien que la définition donnée exclut pratiquement tous les environnements dans lequel évolue un être humain. Cependant, cette classe reste importante pour la robotique industrielle où l'environnement de travail est souvent précisément connu et invariant. Les environnements **dynamiques** contiennent en leur sein des éléments dont les caractéristiques peuvent évoluer au cours du temps. Cela recouvre bien évidemment la présence d'objets mobiles ou non, non identifiés dans la carte disponible. Ce type correspond bien évidemment à la majorité des environnements que peut rencontrer un être humain et donc au sein desquels un robot mobile peut être amené à se déplacer. Cette dynamique va souvent complexifier la localisation (être humain mobile) ou parfois va apporter des informations supplémentaires (porte ouverte/fermée).

Par ailleurs il est aussi important de distinguer les environnements **structurés** de ceux qui ne le sont pas. Les premiers sont par essence les univers artificiels créés par l'Homme composés d'un dédale de pièces plus ou moins grandes, reliées par des couloirs. Cela correspond donc à des environnements au sein desquels il est parfois difficile de se repérer en raison de la similitude de lieux pouvant être rencontrés. Il est donc intéressant pour se localiser de pouvoir identifier des amers caractéristiques, ou de pouvoir utiliser des amers artificiels préalablement introduits dans l'environnement. Les seconds, **non structurés**, correspondent à l'environnement naturel. On manque donc de primitives géométriques régulières pouvant être exploitées pour se localiser. On parle alors d'environnement **semi-structuré**. Évidemment historiquement la localisation en extérieur a toujours été une tâche délicate qui s'est beaucoup simplifiée avec l'avènement du G.P.S. qui procure une localisation précise et absolue.

### 3.3.2 Les modèles d'environnement

En fonction des objectifs poursuivis plusieurs modèles de cartes peuvent être rencontrés. Dans [57] les auteurs proposent une décomposition hiérarchique (Figure 3.9 a) inspirée de [79]), reprise partiellement dans les standards de l'IEEE [107] (Figure 3.9 b)).

A la base on trouve les cartes **métriques** qui permettent de représenter l'environnement comme un ensemble d'éléments (*features*) de position connue dans un espace de référence, généralement 2D. On parle de représentation **explicite** de l'environnement. Nous nous appuyons ici sur les définitions données dans [107]

- Les cartes métriques sont constituées d'une collection d'éléments (points, lignes, cellules, etc.) pour lesquels, si une métrique est définie, il est toujours possible de calculer la

distance les séparant 2 à 2. On peut remarquer que des incertitudes sont prévues comme attribut des coordonnées  $(x, y, \theta)$ . Le plus souvent on utilise des coordonnées cartésiennes dans le repère absolu.

- Les cartes métriques **géométriques** (continues) sont constitués d'une liste d'éléments géométriques caractéristiques (points, lignes, repères, etc.)
- Les **grilles** constituent des cartes discrètes qui décomposent l'environnement en un ensemble de cellules (généralement carrées) constituant des éléments atomiques d'information sur l'environnement. Pour une grille d'occupation cette information correspond justement pour chaque cellule, à sa probabilité d'occupation par un obstacle.

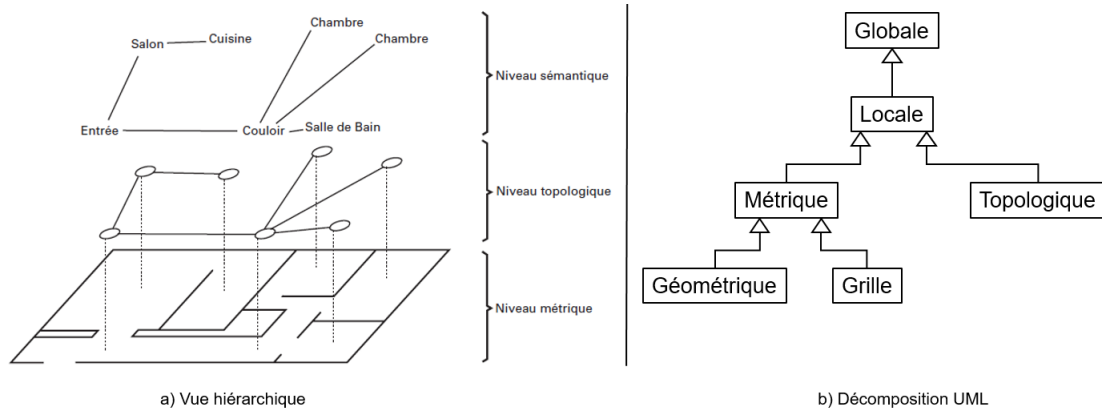


FIGURE 3.9 – Les modèles de représentation de l'environnement

Les principaux avantages des cartes métriques sont qu'elles représentent tout l'environnement connu et qu'elles autorisent une estimation quantifiable et continue de la position du robot si les incertitudes capteurs et la discernabilité de l'environnement le permettent.

Le niveau suivant est constitué des cartes **topologiques** où l'environnement est vu sous la forme d'un graphe, pouvant être hiérarchisé, représentant les relations existants entre différents lieux. On parle de **représentation implicite** de l'environnement. Un exemple illustratif est une carte de métro où les places [57] correspondent aux stations et les arcs aux lignes les reliant. Cette discrétisation de l'environnement constitue un premier avantage car elle simplifie la recherche de chemin qu'il est plus facile de trouver au sein d'un graphe que dans un environnement continu 2D, mais elle n'est exploitable que dans des zones que le robot a déjà parcourues. Par ailleurs, se localiser au sein d'une telle représentation de l'environnement ne nécessite pas nécessairement de disposer d'informations métriques mais d'être capable d'identifier le lieu dans lequel se trouve le robot. C'est donc un avantage certain de ce type de modèle puisqu'il est toujours délicat de transformer des données capteurs en informations géométriques empreintes de nombreuses sources d'erreurs. Il est donc possible de faire appel aux données brutes de perception. Mais évidemment, à contrario, la précision de localisation restera limitée.

Enfin au sommet on trouve une vue **sémantique** de l'environnement où les lieux se déclinent en terme de type de pièce, voire de fonction ou d'objets associés. Ce modèle sémantique, qui ne présente un intérêt que pour le raisonnement abstrait de haut niveau [42] n'est pas considéré dans la décomposition normative proposée dans [107] spécifiquement pour la navigation des robots mobiles (Figure 3.9 b)). Ici les cartes se déclinent en une carte globale pouvant être comme un ensemble de cartes locales, chacune d'elles ne pouvant être que métrique, ou topologique.

Ce paragraphe a présenté les modèles pouvant être utilisés pour modéliser l'environnement. Nous allons maintenant aborder les méthodes permettant au robot de se localiser au sein d'une carte [80].

## 3.4 Les approches de localisation

Sans entrer dans les détails, et sans prétendre être complet, ce paragraphe va permettre de réaliser un tour d’horizon des principales caractéristiques des méthodes de localisation développées en robotique essentiellement pour des cartes métriques. Ces approches pourront ou non composer avec le problème de l’**ambiguïté perceptuelle** (*Perceptual aliasing*) qui fait que plusieurs lieux d’une carte correspondent à une perception identique et ne sont donc pas directement discernables. Nous commencerons par les approches "classiques" appliquées dans le cadre des problèmes de suivi de posture et de localisation globale avant d’aborder d’autres techniques, moins courantes, pouvant aussi y apporter une réponse.

### 3.4.1 Localisation locale : Suivi de posture

#### 3.4.1.1 Méthodes non probabilistes

La terminologie *localisation locale*, utilisée pour le suivi de posture, dénote une démarche continue, privilégiant une vision robot-centrée, effectuant régulièrement de légères corrections de l’estimation de la pose du robot [186].

Dans un premier temps il est possible de n’utiliser que des informations proprioceptives pour mettre à jour la position estimée du robot à partir de celle précédemment calculée (Figure 3.10 a)). On parle alors de **navigation à l’estime** (*Dead-Reckoning*). Dans ce cas, les petites incertitudes sur les déplacements du robot vont se cumuler dans le temps rendant la précision de localisation non bornée. Sans aucune référence à l’environnement, on recueille directement, et généralement à haute fréquence, des informations :

- Directement sur les déplacements élémentaires réalisés en faisant appel au système odométrique du robot [61] [13] ou à de l’odométrie visuelle [172] [147].
- Sur les accélérations observées par les systèmes inertiels, ce qui permettra de remonter (en théorie), par intégration, aux déplacements élémentaires effectués. Cela rend la dérive en position de ces dispositifs d’autant plus importante que les mesures d’accélération sont très bruitées. C’est pourquoi les données de déplacement obtenues pour ce type de dispositif sont souvent fusionnées avec d’autres informations extéroceptives.

Ainsi, la navigation à l’estime ne reste viable que sur des distances modestes. Pour chercher à stopper la croissance de l’incertitude de positionnement du robot une solution est de faire appel, aussi souvent que possible, aux informations extéroceptives (Figure 3.10 b)) pour réduire, de façon absolue dans le repère Monde, l’estimation de la pose du robot. On parle alors de **recalage**. En utilisant de façon combinée une représentation mémorisée locale ou globale de l’environnement avec les informations de perception on va chercher à corriger l’estimée de pose. Plusieurs moyens de recalage absolu peuvent être utilisés :

- L’utilisation de marqueurs (*landmark*) géo-référencés dans l’environnement et donc dans la carte.
  - Ces marqueurs peuvent être **actifs** (*beacon*) et envoyer des signaux (G.P.S., wifi, infra-rouge, bluetooth, etc.) qui seront utilisés par le robot pour se localiser de façon absolue par triangulation (basée angle) [76] ou trilatération (basée distance) [66].
  - Ces marqueurs peuvent aussi être **passifs**. Dans ce cas c’est le robot, au cours de son évolution, qui doit chercher à les détecter. On distingue :
    - ◊ Les marqueurs **artificiels** (QR-Code [144], balise RFID (Radio Frequency Identification) [62], etc.), qui sont introduits volontairement dans l’environnement et

géoréférencés. Ils permettent par exemple d'éviter des situations d'aliasing perceptuel et sont particulièrement utilisés dans les environnements structurés et très difficiles d'emploi en extérieur.

- ◊ Les marqueurs **naturels** qui correspondent à des amers caractéristiques présents dans l'environnement tels que les portes, fenêtres en intérieur ou les routes, arbres et immeubles en extérieur par exemple. Ils sont plus difficiles à identifier et on a donc recours à des techniques de perception et de classification qui ont donné lieu à la proposition de très nombreux algorithmes [185] [209].
- Aux méthodes ancrées sur l'utilisation de marqueurs locaux pour localiser le robot, s'ajoutent des méthodes plus globales orientées cartes où l'on va chercher la meilleure corrélation entre plusieurs cartes. On parle aussi de *model matching*. Dans ce cas on va utiliser les signatures issues des capteurs extéroceptifs de façon à les retrouver au sein d'une carte de l'environnement. Par exemple, pour les cartes métriques, il est possible de localiser le robot en cherchant la meilleure correspondance entre une carte locale dressée à partir des informations capteurs, et la carte globale de l'environnement. Pour y parvenir Olson propose dans [175] une méthode statistique basée sur le principe de maximum de vraisemblance pour comparer les deux cartes employées. L'utilisation des cartes topologiques à des fins de localisation sera abordée plus spécifiquement dans le cadre des méthodes basées sur les apparences.
- Il faut aussi évoquer toutes les techniques actives externes qui permettent de déterminer avec beaucoup de précision la pose d'un robot mobile dans le référentiel Monde (réalité terrain). Dans ce cas, très particulier, ce n'est pas le robot qui détermine lui-même sa position, mais ce sont des dispositifs de localisation externes qui le traquent tout au long de son évolution et peuvent le localiser au sein de son environnement. En intérieur on fait appel à des capteurs actifs comme le système de localisation ViCon [162] qui utilise des caméras infra-rouges à haute résolution avec des marqueurs réfléchissant positionnés sur le robot. Mais évidemment son utilisation est très contraignante, couteuse et d'usage limité. On peut aussi utiliser des capteurs passifs comme des caméras externes classiques localisant un motif particulier présent sur le robot [136]. Ces approches peuvent bien évidemment être intégrées dans le cadre de la problématique de positionnement global.

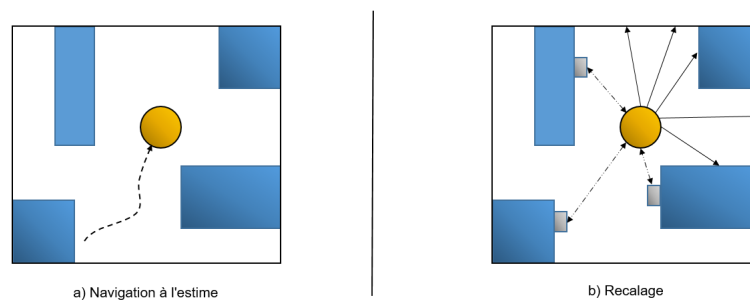


FIGURE 3.10 – Suivi de posture

Les méthodes que nous venons d'évoquer combinent localisation à l'estime avec des opérations de recalage absolu pour réaliser le suivi de posture. Elles combinent donc alternativement des informations proprioceptives avec des informations extéroceptives. On comprend alors l'intérêt de fusionner ces deux sources d'informations. Il semble donc naturel de chercher à combiner en permanence, toutes les données sur le mouvement du robot et sur son environnement qui sont à disposition, à des fréquences différentes, et avec des points de vue pouvant être complémentaires. Cette fusion **multi-capteurs**, qui devrait permettre une localisation plus précise, s'inscrit dans le cadre conceptuel des approches probabilistes que nous allons maintenant présenter.

### 3.4.1.2 Méthodes probabilistes

Comme le note Thrun dans ses travaux [218] [221], les approches probabilistes semblent naturelles pour résoudre la problématique de la localisation en robotique mobile car elles permettent de modéliser les incertitudes liées aux différentes sources de bruit ainsi que leur impact sur l'estimation de la pose du robot.

L'environnement est caractérisé par son état  $x_t$  à l'instant  $t$ . Le robot lui même fait partie de l'environnement et, lorsqu'il se déplace en son sein, il le modifie. En conséquence l'état se traduit sous la forme d'un vecteur dont les composantes peuvent refléter par exemple la pose du robot, son état cinématique et dynamique, mais aussi des informations sur son environnement proche (marqueurs par exemple), etc... Cet état sera toujours incomplet.

Tout au long de son évolution le robot va influencer son environnement et donc l'état de ce dernier par le biais des contrôles qui lui seront appliqués dans le temps pour se déplacer. On note  $u_t$  le contrôle permettant au robot de se déplacer entre les instants  $t-1$  et  $t$ . Ces déplacements élémentaires peuvent aussi s'obtenir grâce au relevé des informations des capteurs proprioceptifs.

Tout au long de son évolution le robot va aussi recueillir des données sur son environnement proche grâce aux capteurs extéroceptifs. On note  $z_t$  les informations relevées à l'instant  $t$ .

En supposant, et c'est une hypothèse forte, que l'état du robot à un instant  $t$  ne dépende que de son état précédent et de la commande courante entre  $t-1$  et  $t$  (Hypothèse Markovienne), alors on peut écrire que la **probabilité de transition d'état** de  $x_{t-1}$  à  $x_t$  est donnée par la loi de distribution de probabilité  $p(x_t|x_{t-1}, u_t)$  où  $p(A|B)$  traduit la probabilité conditionnelle. Cette relation traduit bien implicitement que la pose courante dépend de la pose précédente impactée par le déplacement induit par le contrôle  $u_t$ . Cela définit un modèle de déplacement.

Il est aussi possible de définir la **probabilité de mesure** par l'équation 3.5 qui exprime le fait que l'état courant est suffisant pour permettre de prédire les mesures extéroceptives qui peuvent être effectuées. Cela correspond au modèle de perception suivant où la notation  $v_{t1:t2}$  désigne  $v_{t1}, v_{t1+1}, \dots, v_{t2}$  et représente l'historique d'évolution d'une grandeur  $v$ .

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t) \quad (3.5)$$

Cependant l'état du robot n'est jamais directement accessible et ne peut être inféré qu'à partir des données dont il dispose. Pour y parvenir les méthodes probabilistes s'appuient sur le corpus mathématique de l'**estimation Bayésienne** dans lequel la localisation du robot est décrite de façon probabiliste au travers d'une **croissance** (*Belief* - bel) interne que le robot a de l'état de son environnement et est traduite par une **fonction de densité de probabilité** (*Probability Density Function* - PDF ) balayant tout l'espace.

On peut alors définir deux croyances englobant tout l'historique d'évolution du robot :

- La prédiction (*prior*) :  $bel^-(x_t) = p(x_t|z_{1:t-1}, u_{1:t})$  où les dernières mesures extéroceptives  $z_t$  **ne sont pas encore intégrées** pour calculer la croyance que le robot a de son état.
- La croyance corrigée (*posterior*) :  $bel^+(x_t) = p(x_t|z_{1:t}, u_{1:t})$  où les dernières mesures extéroceptives  $z_t$  **sont intégrées** pour calculer la croyance que le robot a de son état. Le passage de  $bel^-(x_t)$  à  $bel^+(x_t)$  constitue donc une phase de **correction**.

En supposant que l'on travaille sur un espace continu et en utilisant le théorème de la probabilité totale il est possible de démontrer l'Equation 3.6 qui établit une relation de récurrence entre  $x_t$  et  $x_{t-1}$  :

$$bel^-(x_t) = \int p(x_t|u_t, x_{t-1}).bel^+(x_{t-1})dx \quad (3.6)$$



En faisant appel, au Théorème de Bayes il est possible de démontrer la relation 3.7 qui permet de relier  $bel^+$  à  $bel^-$  tout en corrigeant ce dernier en prenant en compte la probabilité d'observer  $z_t$  :

$$bel(x_t) = bel^+(x_t) = \eta \cdot p(z_t|x_t) \cdot bel^-(x_t) dx \quad (3.7)$$

Le filtrage a pour objectif de déterminer des estimateurs des variables d'un système dynamique bruité et partiellement observé. Dès lors, l'utilisation itérative de ces deux équations permet de définir un filtre récursif, fusionnant, dans le temps, les données des capteurs proprioceptifs et extéroceptifs, bruitées et parfois incomplètes, pour déterminer la croyance que peut avoir le robot dans son état, et donc sur sa pose. Ce principe, a donné lieu au développement d'une des techniques d'estimation d'état les plus populaires de la robotique, le **filtre de Kalman** (*Kalman filter*) [123] et ses dérivés. Ce filtre, dans sa version originale, s'appuie sur un ensemble d'hypothèses devant être satisfaites par le système étudié :

1. L'hypothèse Markovienne est supposée satisfaite.
2. Les croyances sont représentées par des Gaussiennes uni-modales de moyenne  $\mu_t$  et de covariance  $\Sigma_t$ .
3. Les bruits de transition d'état et de mesure sont supposés être des bruits blancs, à moyenne nulle, pouvant être représentés par des Gaussiennes.
4. Les modèles de transition d'état et de mesure sont supposés être des modèles linéaires.

La mise en œuvre du filtre de Kalman peut se schématiser par la figure 3.11. On y retrouve la prédiction d'état  $X_t^*$  qui correspond à  $bel^-(x_t)$ , l'étape de correction qui va intégrer les mesures courantes  $z_t$  et une prédiction de ces dernières  $z^*$ , faisant appel à la carte, pour déterminer une estimation de l'état courant  $\hat{X}_t$  qui correspond à  $bel^+(x_t)$ . Ce dernier par récurrence étant réinjecté en entrée du filtre ( $\hat{X}_{t-1}$ ) pour la nouvelle étape de prédiction intégrant les nouvelles informations proprioceptives. Les incertitudes Gaussiennes associées sont représentées par leur covariance  $\Sigma$ .

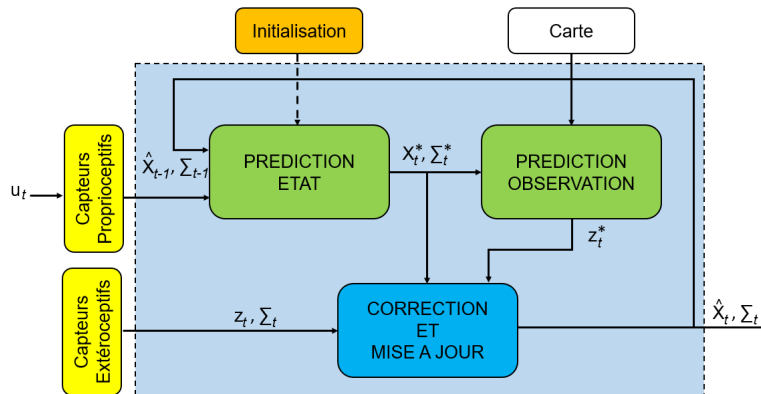


FIGURE 3.11 – Principe du filtre de Kalman

Une des limitations du filtre de Kalman est que les modèles linéaires utilisés sont rarement valides dans la réalité. Une solution naturelle pour pallier cette difficulté est de linéariser les modèles non-linéaires devant être employés en faisant appel à un développement de Taylor du 1<sup>er</sup> ordre au voisinage de la moyenne, ce qui va faire apparaître des matrices Jacobiennes dans les équations manipulées. On parle alors de **filtre de Kalman étendu** (*Extended Kalman Filter*) [127]. Cependant, en l'absence d'une initialisation précise, le filtre peut diverger. Par ailleurs la linéarisation locale peut conduire à des résultats erronés en présence de non linéarités fortes, ou de disparités importantes dans la perception. Dans ce cas les **filtres de Kalman sans parfum** (*Unscented Kalman Filter*) [121] proposent d'utiliser un nuage de points pour

déterminer les paramètres (moyenne, variance) de la Gaussienne de transformation, ce qui permet une meilleure estimation d'état et d'observation.

Les filtres uni-modaux que nous venons de décrire ne permettent de suivre qu'une seule hypothèse d'état (*Single Hypothesis Tracking* - SHT). A partir d'une position initiale assez bien connue, ils peuvent donc répondre au problème de suivi de posture en l'absence d'aliasing perceptuel. Pour pouvoir appliquer ce type d'approche dans des environnements présentant une plus forte variabilité, il semble donc naturel, dans un premier temps, d'étendre les filtres de Kalman pour pouvoir traquer simultanément plusieurs hypothèses d'états (*Multiple Hypothesis Tracking* - MHT) [120] [200], au prix d'un important coût de calcul. Dans ce cas là, on va constituer des banques de filtres de Kalman, suivant chacun une hypothèse, dont la crédibilité va varier au gré des informations proprioceptives et extéroceptives recueillies. A un instant donné, le plus plausible sera retenu, mais on garde, à chaque fois, la trace de ceux qui ont été écartés. Afin de limiter la croissance exponentielle des hypothèses à prendre en compte, celles voisines sont regroupées. Cependant, même si cette approche reste bien plus robuste que le SHT, rien n'assure que l'état du système fasse bien partie de ceux traqués.

Les filtres Gaussiens que nous venons de balayer sont performants mais souffrent encore de deux principales limitations. La première, et non des moindres, est que les incertitudes sont toujours modélisées par le biais d'une, ou d'une somme de lois Gaussiennes. Cela n'est pas toujours vérifié. Par ailleurs, ils peinent à adresser globalement la problématique de localisation, au profit d'un suivi privilégié local de certaines hypothèses de localisation. Pour répondre à ces limitations nous allons maintenant présenter les filtres **non-paramétriques** particulièrement bien adaptés pour représenter les croyances multimodales complexes.

### 3.4.2 Localisation globale

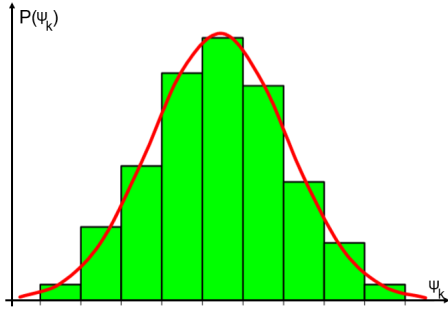
A contrario de l'estimation locale, l'estimation globale s'adresse à des méthodes discontinues, appréhendant l'environnement connu dans sa globalité et pouvant effectuer, si nécessaire, des corrections importantes sur l'estimée du robot [186]. Les méthodes que nous allons présenter s'inscrivent dans le cadre général du filtrage Bayésien, pour lequel on va chercher à évaluer, en balayant le plus largement possible l'espace d'état envisageable (la carte), la croyance  $bel^+(x)$  que l'on peut avoir sur la pose du robot, et cela, toujours en s'appuyant sur l'intégration des informations proprioceptives puis extéroceptives, sans pour autant gérer explicitement le suivi des hypothèses comme les méthodes de suivi de posture.

#### 3.4.2.1 Localisation par grille

Dans ce type d'approche, dite Markovienne, l'espace est décomposé en une grille d'un ensemble fini de  $K$  cases  $\psi_i$ . L'objectif de la localisation est alors de déterminer, à l'instant  $t$ , la distribution de probabilité de présence du robot, à travers tout l'espace discrétisé. Il faut remarquer ici que, à l'intérieur d'une case, la densité de probabilité de présence est supposée constante. Dans ce cadre d'étude, la répartition probabiliste continue qui était utilisée pour le filtrage de Kalman, fait place à une représentation approchée normalisée, induite par la discrétisation de l'environnement, correspondant à un histogramme (Figure 3.12 a).

Dès lors l'algorithme mis en œuvre (Algorithme 1) est une discrétisation de l'algorithme général de filtrage Bayésien.  $x_t$  y représente une variable aléatoire associée à l'état du robot à l'instant  $t$ . En entrée on connaît évidemment la distribution discrète de probabilité au travers des  $K$  cellules de la grille à l'instant  $t-1$ , ainsi que les informations proprioceptives induites et/ou la commande  $u_t$  appliquée entre  $t-1$  et  $t$ , les données extéroceptives  $z_t$  et bien sûr la carte  $m$ . Lors de l'initialisation, ne connaissant pas la posture initiale, chaque cellule se voit attribuer la même

probabilité ( $\frac{1}{K}$ ). Lors de chaque itération de l'algorithme, celui-ci va mettre à jour la distribution de probabilité de présence du robot dans chacune des cellules en la décomposant classiquement en 2 phases. D'une part la prédiction (ligne 2) qui permet de calculer la probabilité  $p_{k,t}^-$  de se retrouver dans une case en partant d'une autre case connaissant le modèle de déplacement. D'autre part la phase de correction (ligne 3) qui va mettre à jour cette probabilité en prenant en compte les informations extéroceptives recueillies, les modèles de perception, et la carte, pour obtenir finalement  $p_{k,t}$ .



a) Histogramme d'une Gaussienne

---

**Algorithme 1** : Localisation par grille inspiré de [221]

---

**Input** :  $\{p_{k,t-1}\}_{k=1,\dots,K}, u_t, z_t, m$

1 **for**  $k = 1$  **to**  $K$  **do**

2      $p_{k,t}^- = \sum_{i=1}^K p(x_t = \psi_k | x_{t-1} = \psi_i, u_t, m) \cdot p_{i,t-1}$

3      $p_{k,t} = p(z_t | x_t = \psi_k, m) \cdot p_{k,t}^-$

4 **end**

**Output** :  $\{p_{k,t}\}_{k=1,\dots,K}$

---

FIGURE 3.12 – Localisation par grille

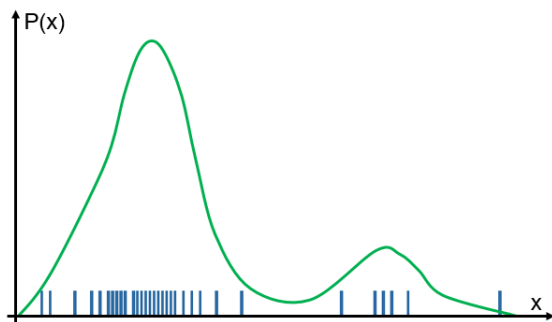
Cette méthode [47] permet de considérer des distributions de probabilité multimodales et d'adresser, à tout instant, la problématique de localisation absolue à travers tout l'espace connu représenté par une grille. Elle a même été étendue aux environnements dynamiques en faisant appel à des techniques de filtrage [84]. Sa principale limitation est que sa précision est directement fonction de la taille des cellules de la grille. Sans compter le fait que si la position peut simplement être représentée en deux dimensions, la prise en compte de l'orientation conduit à intégrer une nouvelle dimension, multipliant par la même le nombre de grilles à prendre en considération. Cela rend cette approche plutôt adaptée aux environnements de dimensions modestes [184]. La complexité algorithmique de ce type d'approche va donc être très gourmande tant en temps de calcul, qu'en espace mémoire. Pour y faire face, des techniques de sous-échantillonnage des informations capteurs et de mise à jour sélectives peuvent être employées.

### 3.4.2.2 Filtrage particulaire

La méthode de localisation par grille traque donc le robot à travers l'ensemble de l'environnement. Pour éviter la discrétisation de la carte, une façon alternative, et, à la base, moins coûteuse en temps et en espace est de ne chercher à suivre simultanément que les hypothèses les plus plausibles. Pour ce faire on va représenter la croyance  $bel(x_t)$  que le robot a sur sa posture à un instant donné  $t$  par un ensemble  $M$  de particules  $\chi_t = \{x_t^{[1]}, \dots, x_t^{[M]}\}$ . Chacune de ces particules, initialement aléatoirement distribuée au sein de l'environnement, va se voir associer un poids représentatif de la vraisemblance de la pose correspondante. Cet échantillonnage dit de **Monte Carlo**, permet d'approcher la distribution continue correspondante (Figure 3.13 a)). A partir de cette représentation de la pose du robot dans son environnement, on développe un filtre Bayésien non paramétrique appelé **filtre particulaire** décrit par l'algorithme 2. Il peut se décomposer en 3 principales phases :

1. Prédiction : La ligne 2 permet d'actualiser la position de chaque particule en fonction de la commande appliquée. Une fois toutes les particules traitées  $\{x_t\}$  peut être assimilé à  $bel^-(x_t)$ .

2. Correction : La ligne 3 permet de déterminer le poids associé à chacune des particules en fonction des dernières informations extéroceptives. L'ensemble  $\chi_{temp}$  rassemble alors le nouvel état de chacune des particules et le poids associé (ligne 4).
3. Ré-échantillonnage : Cette étape n'est pas présente dans les filtres Bayésiens classiques. Elle permet d'éviter la dégénérescence du filtre particulaire où, de fait, plus qu'une seule hypothèse de localisation ne resterait probable au fil du temps. Pour ce faire on va ré-échantillonner l'ensemble des particules de façon à privilégier celles des régions où la localisation du robot semble la plus plausible (ligne 7). On réactualise ensuite le poids associé en conséquence (ligne 8). Le nouvel ensemble  $\chi_t$  ainsi créé constitue alors une approximation de  $bel(x_t)$  (ligne 9).



a) Échantillonnage de Monte Carlo

---

**Algorithme 2** : Localisation par filtre particulaire inspiré de [221]

---

**Input** :  $\{\chi_{t-1}, u_t, z_t\}$

```

1  $\chi_{temp} = \chi_t = \emptyset$ 
  for  $k = 1$  to  $M$  do
2    $x_t^{[k]} \sim p(x_t | u_t, x_{t-1}^{[k]})$ 
3    $w_t^{[k]} = p(z_t | x_t^{[k]})$ 
4    $\chi_{temp} = \chi_{temp} + \langle x_t^{[k]}, w_t^{[k]} \rangle$ 
5 end
6 for  $k = 1$  to  $M$  do
7    $x_t^{[k]} = MCL(\chi_{temp})$ 
8    $w_t^{[k]} = p(z_t | x_t^{[k]})$ 
9    $\chi_t = \chi_t + \langle x_t^{[k]}, w_t^{[k]} \rangle$ 
10 end
Output :  $\chi_t$ 

```

---

FIGURE 3.13 – Localisation par filtrage particulaire

De nombreuses améliorations ont été apportées à l'algorithme de principe présenté ici en s'intéressant à plusieurs points clés affectant la qualité de l'estimation que sont le choix de la fonction d'échantillonnage [206], ou du nombre de particules [211] afin d'arriver à un compromis entre robustesse, précision, et rapidité de calcul.

Cette approche particulière de la localisation, encore appelée localisation de Monte Carlo, peut prendre en compte des distributions multimodales complexes et les systèmes non linéaires. Elle a donc largement été employée en robotique [217] pour adresser les problématiques de localisation globale [54] [240] et de kidnapping [46].

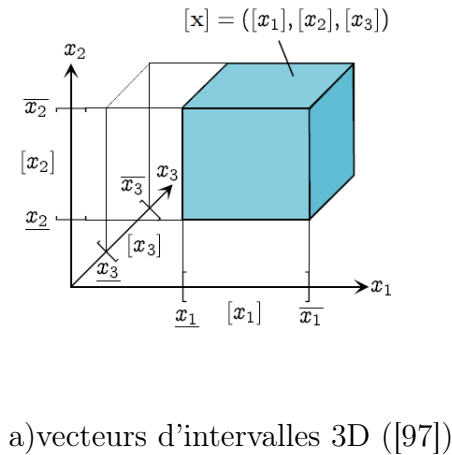
### 3.4.3 Autres approches

#### 3.4.3.1 Méthodes ensemblistes

Si les méthodes probabilistes approchent la problématique de localisation au travers d'une vision faisant appel à la notion de probabilité, les méthodes ensemblistes que nous allons maintenant présenter l'abordent en adoptant sous l'angle du bornage. Ce point de vue semble donc particulièrement intéressant puisqu'il fournit, à un instant donné, une garantie sur la localisation du robot à l'intérieur des limites identifiées. On peut cependant remarquer que, dans ce contexte, la notion de pose n'a pas de sens en tant que tel, puisque qu'aucune position ou orientation n'est privilégiée à l'intérieur d'une boîte.

Dans ce type d'approche, sous réserve que les incertitudes et erreurs soient bornées, les données manipulées sont représentées par des intervalles  $[x]$  correspondant à un sous-ensemble connexe, fermé, de  $\mathbb{R}$ . Un intervalle  $[x]$  est alors caractérisé par une borne inférieure  $\underline{x}$ , supérieure  $\bar{x}$ , un centre  $mid([x])$  et une taille  $w([x])$ . Les opérations ensemblistes classiques peuvent s'appliquer (intersection, union) sur les intervalles. Il est alors possible de définir un **vecteur d'intervalles** (appelé *boite* ou *pavé*) comme un sous-ensemble de  $\mathbb{R}^n$  étant le produit cartésien de  $n$  intervalles (Équation 3.8). La Figure 3.14 illustre la notion de vecteur d'intervalles pour un espace à 3 dimensions. Les opérations ensemblistes d'intersection et d'union peuvent être étendues aux vecteurs d'intervalles et l'on peut approcher un pavé  $\mathbb{X}$  de  $\mathbb{R}^n$  par un sur-pavage  $\mathbb{X}^+$  et sous-pavage  $\mathbb{X}^-$ . On a alors  $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$ . L'arithmétique des intervalles présente l'intérêt de pouvoir aisément transposer les opérations classiques pouvant être appliquées aux nombres réels (+, -, \*, /, relations d'ordre total ou partiel) ainsi que les fonctions élémentaires (sin, cos, exp, etc..). Si l'on connaît les contraintes pouvant s'appliquer sur certaines variables il est possible de définir des **contracteurs** de façon à limiter la dimension des pavés.

$$[x] = [x_1] \times \dots \times [x_n] = ([x_1], \dots, [x_n]) \quad (3.8)$$



a) vecteurs d'intervalles 3D ([97])

---

### Algorithme 3 : SIVIA

---

```

Input :  $\{[x], f, \mathbb{Y}, \epsilon, \mathbb{X}^-, \mathbb{X}^+\}$ 
Output :  $\{\mathbb{X}^-, \mathbb{X}^+\}$ 
1 if ( $[f]([x]) \cap \mathbb{Y} = \emptyset$ ) then
2   return
3 else if ( $[f]([x]) \subset \mathbb{Y}$ ) then
4   |  $\mathbb{X}^- := \mathbb{X}^- \cup [x]$ 
5   |  $\mathbb{X}^+ := \mathbb{X}^+ \cup [x]$ 
6 end
7 else if ( $w([x]) < \epsilon$ ) then
8   |  $\mathbb{X}^+ := \mathbb{X}^+ \cup [x]$ 
9 end
10 else
11   |  $([x_1], [x_2]) = \text{Bissecter}([x])$ 
12   |  $(\mathbb{X}^-, \mathbb{X}^+) = \text{SIVIA}([x_1], f, \mathbb{Y}, \epsilon, \mathbb{X}^-, \mathbb{X}^+)$ 
13   |  $(\mathbb{X}^-, \mathbb{X}^+) = \text{SIVIA}([x_2], f, \mathbb{Y}, \epsilon, \mathbb{X}^-, \mathbb{X}^+)$ 
14 end
15 end

```

---

FIGURE 3.14 – Algèbre ensembliste par intervalles

L'efficacité des méthodes ensemblistes repose sur celles des algorithmes d'inversion ensemblistes. SIVIA (*Set Inversion via Interval Analysis*)[119] (Algorithme 3) est un algorithme *Branch and Bound* récursif permettant de caractériser un pavage  $\mathbb{X}$  satisfaisant l'équation  $f(\mathbb{X}) = \mathbb{Y}$  en résolvant l'équation  $\mathbb{X} = f^{-1}(\mathbb{Y})$ . Il permet de déterminer un sous et un sur-pavage de  $\mathbb{X}$ . Dans le cas de la localisation  $\mathbb{Y}$  représente les mesures réalisées, la fonction  $f$  le modèle reliant les mesures à une pose, et enfin  $\mathbb{X}$  le pavé à l'intérieur duquel doit se trouver le robot compte tenu des premiers éléments.

Le problème de localisation ensembliste peut alors être formulé comme un problème de propagation de contraintes (*Constraints Satisfaction Problem*). Cependant cette approche permettant d'apporter une garantie reste très sensible aux données aberrantes (*outlier*) qui se

rencontrent fréquemment lors d'expérimentations. Pour répondre à cette problématique l'algorithme robuste RSIVIA (*Robust SIVIA*) [118] qui permet de tolérer la présence de  $q$  mesures erronées au milieu de  $m$  observations a été proposé.

Ce type d'approche a été largement utilisé en robotique. Le problème de suivi de posture, par exemple, peut être formulé comme pour les filtres Bayésiens sous la forme de l'Équation 3.9 où  $[x_t]$  et  $[u_t]$  représentent les pavés associés à l'état  $x_t$  et à la commande  $u_t$  du robot à l'instant  $t$ ,  $[f]$  représente une fonction d'inclusion caractérisant la dynamique du robot. Sans l'intégration des observations les dimensions du pavé associé à  $x_t$  vont s'accroître avec le déplacement du robot. Les pavés des données extéroceptives  $[z_t]$  sont donc utilisés pour recalibrer la position du robot et diminuer la taille de  $[x_t]$ . On peut remarquer que, de fait, en apportant une réponse garantie, les approches ensemblistes effectuent du MHT.

$$[x_{t+1}] = [f]([x_t], [u_t]) \quad (3.9)$$

Ce principe a été utilisé par exemple en intérieur dans [128] pour localiser un robot doté d'ultrasons, ou en extérieur dans [116] où la localisation d'un robot sous-marin est mise en œuvre dans un contexte de SLAM. Cependant l'opération de recalage est effectuée hors ligne. Dans [90] la localisation ensembliste d'une voiture dotée d'odomètre et d'un gyromètre est comparée avec celle obtenue à partir d'un filtre de Kalman étendu. Les résultats donnent évidemment une précision moindre, mais acceptable et garantie. Dans [98] l'algorithme LUVIA (*Localisation Updating with Visibility and Interval Analysis*) propose d'intégrer, dans un contexte multi-robots, une information sur la visibilité inter-robots pour compenser la dérive odométrique.

La problématique de localisation globale où la pose initiale n'est pas connue a bien évidemment aussi été abordée. [99] propose une méthode ensembliste où seules les données laser, d'odométrie et une grille d'occupation de l'environnement permettent de déterminer la pose d'un robot terrestre sans connaître sa position initiale. Elle montre des résultats comparables avec une approche de Monte-Carlo. Cependant un des problèmes majeur des approches ensemblistes est de correctement estimer les limites des bruits et de mesures aberrantes. Une sous-estimation peut conduire à une absence de solution lors de l'opération de contraction, et une sur-estimation à des valeurs trop pessimistes (pavés de grandes dimensions). Ce problème est traité dans [64] grâce à l'algorithme BEES qui permet de prendre en compte des événements perturbateurs tels que les mesures aberrantes ou le kidnapping. Des méthodes hybrides, couplant méthodes probabilistes et ensemblistes, telles que [17] (Filtre de Kalman et IAL), ou [171] (Monte-Carlo et IAL), ont aussi été proposées.

En conclusion, les approches ensemblistes constituent donc une alternative crédible aux approches probabilistes en lui adjoignant la notion de garantie.

### 3.4.3.2 Méthodes basées sur les apparences

Les méthodes basées sur les apparences (*Appearance based*) sont particulièrement adaptées pour se localiser au sein de cartes topologiques qui se transforment alors en cartes des apparences (*Appearance map*). Leur principe est de chercher à identifier un lieu (un noeud du graphe) uniquement à partir d'informations brutes issues des capteurs extéroceptifs (mesure planaire de distance, image) que l'on peut assimiler à une signature spécifique, en l'absence de similitude de signature évidemment. Par exemple dans [203] les auteurs développent une approche de navigation en extérieure basée sur les apparences utilisant un système de vision. Ils font appel à une représentation hybride topologique-géométrique. Une phase d'apprentissage permet de recueillir et d'organiser en graphe un ensemble d'images-clefs (*Key-image*) et parallè-

lement des points caractéristiques de correspondance sont identifiés dans chaque paire d'images voisines. Lors de la phase de navigation les points caractéristiques extraits en temps réel et leur correspondance dans les images clefs sont utilisés pour contrôler le robot. L'estimation de la précision de localisation n'est pas l'objectif de ces travaux.

Ce type de méthode est aussi très utilisé pour la localisation visuelle au long terme en extérieur et particulièrement pour le problème de fermeture de boucle (*Loop closure*) rencontré en raison de l'inévitable dérive de localisation lorsque l'on effectue du SLAM. Des approches telles que [16] ou [125] cherchent à identifier des images relevées en temps réel avec une banque d'images de référence géolocalisées. Pour ce faire elles proposent des descripteurs performants (Descripteur binaire / Bag-of-Words) permettant de rapidement procéder à l'identification et donc à la localisation. Ces approches qui tiennent plus de la reconnaissance de situation que de la localisation proprement dite doivent gérer l'accroissement de la complexité calculatoire qui croît avec la longueur de la séquence d'acquisition. Ces méthodes doivent particulièrement être insensibles à la variabilité saisonnière et d'éclairage.

On peut aussi chercher à enregistrer les signatures capteurs depuis différentes positions voisines de l'espace. Ainsi, on peut en utilisant le cheminement dual, à partir de l'identification des signatures remonter jusqu'à la pose du robot et même envisager effectuer du suivi de posture. Ce type de démarche reste cependant sensible aux bruits capteurs, à la dynamique de l'environnement, et ne peut discerner des signatures similaires. Cela rend les méthodes basées sur les apparences aussi applicables pour des cartes métriques. Dans [228] l'environnement est décomposé en une grille mais les informations de localisation ne sont pas mémorisées sous forme de données géométriques mais d'une représentation statistique de l'environnement. On discrétise donc l'espace avec un grain bien plus fin et régulier que pour les cartes topologiques. En créant une carte composée de relevés représentatifs de la position du robot dans la carte, il est alors possible en comparant la signature laser courante à celles mémorisées de déterminer par cross-corrélation la position et l'orientation courante absolue du robot. Plus récemment dans [188] ce type d'approche a été utilisé, pour un robot de service, au sein du système de localisation actif basé sur une approche probabiliste classique, en utilisant des images stéréoscopiques intégrant des informations de profondeur.

### 3.4.4 Conclusion

Cette partie a permis d'effectuer un tour d'horizon des méthodes de localisation utilisées en robotique pour répondre aux problèmes du suivi de posture et de la localisation globale. Nous en avons présenté les grands principes, ainsi que les principales limitations. À la vue de la grande diversité des approches proposées et de la quantité colossale de publications relevant de ce domaine on peut, sans se tromper, avancer que la problématique de la localisation est centrale pour la robotique mobile, et nous n'avons volontairement pas abordé tous les travaux relevant du SLAM qui sont fort nombreux actuellement.

Tous ces travaux essaient d'exploiter au mieux, au cours du temps, les informations à priori (carte), internes (commandes et données proprioceptives), ou externes (extéroceptives) que le robot peut connaître ou collecter au cours de son évolution. Ils cherchent à obtenir la meilleure localisation possible, tout en préservant une complexité de calcul compatible avec une utilisation embarquée. La fusion des données capteurs s'impose dans la plupart des travaux pour tenter de limiter, autant que faire ce peut, une dérive de la précision de localisation. En effet les incertitudes de tous ordres entachent tant les modèles utilisés, que les mesures effectuées, et les ordres appliqués au robot.

C'est pourquoi, le prochain paragraphe, après avoir précisé le vocabulaire attaché au concept d'incertitude, va d'une part nous permettre d'identifier plus précisément les sources d'erreurs impactant le processus de localisation et, d'autre part, proposer une analyse critique des précisions de localisation affichées par un ensemble de travaux représentatifs.

## 3.5 La localisation : Retour sur les incertitudes

Dans cette partie nous allons tout d'abord rappeler un ensemble de définitions associées à la notion de mesure, aux erreurs et aux propriétés pouvant y être associées. Nous balaierons ensuite, dans le cadre de la localisation en robotique mobile, les principales sources d'incertitudes pouvant affecter, plus ou moins fortement, la précision de localisation. Enfin, nous analyserons un ensemble de travaux représentatifs des approches de localisation en robotique mobile, en appréciant la précision avancée, aux regards des conditions expérimentales mises en œuvre, et des sources d'incertitudes prises, ou non en compte, et caractérisées.

### 3.5.1 Mesure et incertitude

Une mesure dépend de nombre de facteurs dont, entre autres, du processus expérimental, des conditions expérimentales et de la qualité du capteur l'effectuant. L'erreur de mesure (Équation 3.10) peut être définie de la façon suivante :

$$\hat{X} = X + \tilde{X} \quad (3.10)$$

$\hat{X}$  : Représente la valeur mesurée  
 $X$  : Représente la valeur vraie inaccessible  
 $\tilde{X}$  : Représente l'erreur de mesure

L'erreur correspond donc à la différence entre la valeur mesurée et la valeur vraie de la grandeur mesurée.

On distingue les erreurs **systematiques** dues à des problèmes d'étalonnage, des erreurs **aléatoires**, non reproductibles qui ne peuvent être corrigées et nécessitent un traitement statistique. On a donc la relation (Équation 3.11) :

$$\hat{X} = X + \text{Erreur Systematique} + \text{Erreur Aléatoire} \quad (3.11)$$

Donc une mesure peut être associée à un doublet  $(x, \delta x)$  vérifiant l'Équation 3.12 :

$$x - \delta x \leq X \leq x + \delta x \quad (3.12)$$

$x$  : Représente la valeur mesurée  
 $X$  : Représente la valeur vraie inaccessible  
 $\delta x$  : Représente l'incertitude de mesure

L'**incertitude de mesure** (*uncertainty*)  $\delta x$  correspond à un paramètre positif d'évaluation de la dimension aléatoire d'une mesure. Elle définit un intervalle, autour de la valeur mesurée et corrigée de ses biais, au sein duquel est supposée appartenir la valeur vraie.

Sur une plage de mesures donnée, un très large ensemble de propriétés [6] [95], essentiellement statiques, permet de caractériser la qualité d'un capteur :



- **Exactitude / accuracy** : Degré de concordance entre une valeur mesurée et la valeur vraie.
- **Précision** : Erreur maximale ( $\pm$ ) pouvant être commise sur une mesure.
- **Résolution** : Plus petite différence entre 2 mesures.

A ces propriétés classiques s'ajoutent :

- **Fidélité** : Aptitude d'un capteur à produire les mêmes mesures de façon répétée dans les mêmes conditions expérimentales. C'est un paramètre statistique caractérisé par une variance  $\sigma$  et un écart-type  $\sigma^2$ .
- **Justesse** : Étroitesse de l'accord entre la moyenne d'un nombre infini de valeurs mesurées répétées et une valeur de référence. Elle est caractérisée par un **biais**.
- **Répétabilité** : Valeur minimale de la fidélité pour des mesures faites dans les mêmes conditions expérimentales.
- **Reproductibilité** : Valeur minimale de la fidélité pour des mesures faites dans des conditions expérimentales différentes.

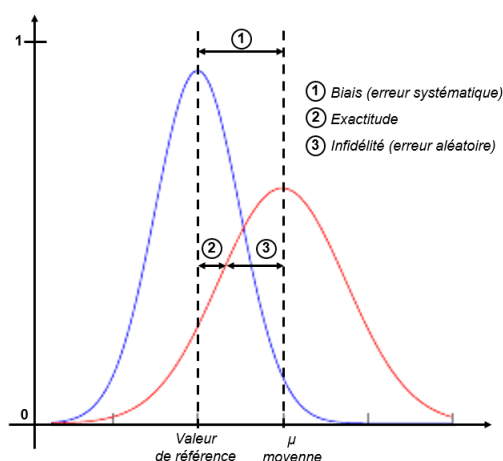


FIGURE 3.15 – Mesures : Illustration de quelques propriétés

### 3.5.2 Les incertitudes : De multiples origines

Pour effectuer un tour d'horizon des incertitudes pouvant impacter le processus d'évaluation de la pose d'un robot mobile nous allons nous appuyer sur la décomposition proposée par Thrun dans [221].

L'**environnement** de la mission est, par essence, la première source d'incertitude. Qui peut prédire ce qui se passera dans le futur ? Bien évidemment, en robotique industrielle par exemple, on va chercher à travailler dans des environnements statiques limitant (mais ne supprimant pas) l'occurrence d'événements inopinés. Mais en robotique mobile, la dynamique de l'environnement, que ce soit en intérieur, ou en extérieur, et donc la présence d'obstacles, et/ou d'événements non planifiés durant la mission est bien évidemment un paramètre incontournable et difficile à anticiper. En revanche, si l'avenir n'est pas prédictible, il est tout de même possible de s'y préparer en mettant en œuvre, au niveau du robot des stratégies qui permettront au mieux de répondre, si nécessaire, à certains événements perturbateurs (comme la méthodologie PANORAMA). En environnement intérieur, il est aussi envisageable d'instrumenter l'environnement dans l'objectif d'assurer un certain niveau de garantie de localisation.

Les **capteurs** sont clairement l'une des sources principales d'incertitudes pour la localisation puisque c'est par leur entremise que le robot peut estimer tant l'évolution de son état, que son évolution au sein de l'environnement. Deux principaux facteurs sont à considérer. D'une part la qualité et les caractéristiques propres du capteur au travers de l'ensemble des propriétés permettant de caractériser la précision de ses mesures. Mais ces performances sont aussi directement liées aux principes physiques sous-tendant des mesures qui sont souvent très dépendantes des conditions expérimentales et donc de l'environnement dans lequel évolue le robot. Cela induit un **bruit de perception** qui traduit la variabilité de mesure qui peut être observée au niveau d'un capteur en fonction des caractéristiques de l'environnement dans lequel il évolue (luminosité, couleur, type matériaux, surface), ou de la qualité des informations traitées (floutage d'image liée aux vibrations). D'autre part, et c'est une des difficultés de l'expérimentation, les capteurs sont soumis à des bruits de mesure, souvent imprédictibles, même si l'on peut espérer qu'ils soient vraiment aléatoires et donc de moyenne nulle, qui viennent noyer l'information utile qui doit être extraite de la mesure.

Les **modèles** utilisés sont eux aussi par définition imprécis puisqu'ils ne sont qu'une approximation, plus ou moins bonne, du monde réel que l'on veut mettre en équation. A titre d'exemple une carte métrique ainsi que les amers qui peuvent s'y trouver, ne sont qu'un reflet, plus ou moins précis, de la vérité terrain. En robotique mobile, les modèles de déplacement utilisés sont souvent des modèles cinématiques, alors que le modèle dynamique, plus complet n'est pas utilisé en raison de la complexité de modélisation du contact roue/sol. Enfin les modèles de mesures peinent à prendre en considération tous les phénomènes physiques qui peuvent venir entacher l'obtention d'une mesure.

L'implantation **logicielle** de la localisation est aussi une source importante d'erreur. En effet, si le monde réel est par nature continu, la mise en œuvre embarquée de la séquence de Navigation-Guidage-Contrôle induit une nécessaire discrétisation tant de l'obtention des données capteurs, que de la génération de la commande. Cette dimension logicielle induit de fait une incertitude de localisation puisque, le robot étant normalement en perpétuelle évolution, on traite des informations passées, pour en déduire avec un peu de retard les ordres à appliquer. Sans compter que les délais de mise à disposition de certaines informations (perception, localisation, commande) peuvent parfois être significatifs.

Enfin le **robot** lui même est source d'incertitudes. En effet d'une part nous ne connaissons que de façon imparfaite ses caractéristiques dimensionnelles propres, ainsi que celles relatives à la position des capteurs qui ont pu lui être ajoutés. Cela peut introduire des imprécisions importantes dans l'estimation de sa pose. D'autre part, nous ne contrôlons que de façon imparfaite les moteurs et donc le déplacement engendré. Sans compter que leur usure peut modifier leur comportement et de façon totalement irrégulière. Par ailleurs il existe aussi un **bruit d'actionnement** qui correspond au fait, que la même action conduite depuis un état connu n'aura pas toujours le même effet. Cela peut provenir d'une modélisation imparfaite de son environnement (glissement, pente, etc.), de forces externes pouvant être appliquées au robot de façon impromptue, et des modèles de déplacement imparfaits utilisés.

Comme on peut le constater ces sources d'incertitudes ne sont pas indépendantes mais souvent corrélées entre elles, et la classification proposée met en avant le rôle central des capteurs en les excluant du robot proprement dit. Dans la suite de ce document nous ne prétendons pas toutes les prendre en compte mais nous chercherons à caractériser les plus significatives. Nous distinguerons les sources d'incertitudes suivantes (Figure 3.16) pour estimer la précision de localisation d'un robot :

- **Incertitude De Carte (IDC)** (*Environnement - Modèle*) : Incertitude liée à notre connaissance imparfaite de la carte métrique de l’environnement, des points de repère géo-référencés qui peuvent s’y trouver, et la localisation spatiale des attributs associés (couleur, matériaux, etc.).
- **Incertitude de Mesure Capteur (IMC)** (*Environnement - Capteur - Modèle*) : Incertitude liée uniquement aux mesures capteur dans le repère capteur.
- **Incertitude de Conception Mécanique (ICM)** (*Robot - Modèle*) : Incertitude liée à notre connaissance, parfois approximative, de la position des repères capteurs par rapport au repère Robot.
- **Incertitude de Traitement Logiciel (ITL)** (*Logiciel - Robot (incluant Capteur)*) : Incertitude liée à l’implantation logicielle de la boucle de Navigation - Guidage - Contrôle.
- **Incertitude de Méthode de Localisation IML** (*Modèle*) : Incertitude minimale théorique pouvant être obtenue par une méthode de localisation.

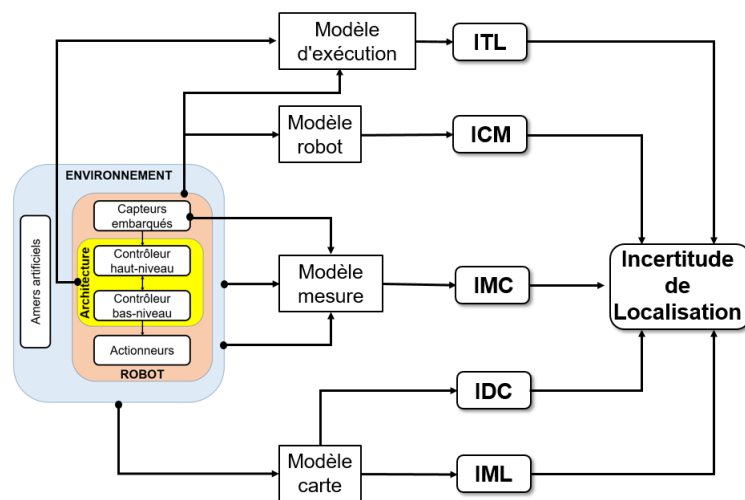


FIGURE 3.16 – Localisation : Sources d’incertitudes

Les sources d’incertitudes de localisation ayant été identifiées et classifiées, nous allons maintenant, au travers de l’étude de quelques articles du domaine balayant les différentes approches utilisées en robotique, analyser, lorsque cela est possible, la précision de localisation affichée, au regard des incertitudes prises en compte.

### 3.5.3 Méthodes de localisation : incertitudes affichées

L’étude que nous allons réaliser balaie une cinquantaine d’articles assez récents, couvrant les principales classes de méthodes de localisation que nous avons présentées, pour lesquels nous avons effectué une analyse critique des performances affichées et de leur contexte d’obtention. Nous nous appuyerons sur les Tables simplifiées 3.1, 3.2 et 3.3. Le lecteur pourra se rapporter aux tableaux détaillés présentés en Annexe B de ce document pour trouver des informations plus complètes intégrant, entre autres, l’évaluation quantitative affichée des performances de Localisation mais aussi d’autres références. Nous scinderons notre analyse selon 2 principales parties. Dans la première nous décomposerons l’étude des articles choisis selon la classe des méthodes de localisation étudiées, à savoir les méthodes locales (probabilistes ou non), les méthodes globales (grille et filtre particulière) pour terminer par les méthodes ensemblistes et topologiques. Dans la seconde nous tirerons des enseignements plus globaux sur les performances affichées et leur contexte expérimental.

### 3.5.3.1 Incertitudes affichées et contexte expérimental

Les critères suivants ont été adoptés pour guider notre discussion :

- La classe des méthodes (**Méthodes**) de localisation utilisées en distinguant celle proposée de celles employées à titre de comparaison.
- La nature des performances (**Perf.**) analysées dans l'article étudié (X, Y,  $\Theta$ , Confiance, Temps de calcul, Vitesse de convergence de l'algorithme de localisation, Erreur relative). On trouvera en B les valeurs affichées.
- La nature des expérimentations (**Test**) effectuées (Simulation (Simu), Expérimentation (Expe.), Inconnue (?)).
- Le type de trajectoire (**Traj.**) employé pour évaluer la méthode étudiée (Droite, Carrée, en S, Complexe).
- L'environnement (**Env.**) choisi pour effectuer les expérimentations / simulations (Complexité, Régularité, Intérieur, Extérieur, etc.).
- La façon dont la réalité terrain (**R.T.**) est obtenue (Capteur externe, amer externe, point de passage, Inconnue (?)).
- Le type de contrôle (**Ctrl**) (Téléopération, Boucle ouverte, Boucle fermée, Inconnu (?)).

Source	Méthodes	Perf.	Test	Traj.	Env.	R.T.	Ctrl
[214]	Fusion Odo/Gyro (proposée)	Distance	Exp.	Carrée	Ouvert Terrain difficile	Laser  Externe	Téléop.
	Odo						
	Gyro						
	EKF-GPS						
[19]	Recalage multi-amers et EKF	Distance Confiance	Exp.	Complexe	Extérieur Chemins goudronnés	?	?
[78]	Détection d'amers (proposé)	Distance	Exp. Hors ligne	Droite	Couloir	Points de passage	?
	CARMEN [167]						
[20]	Odo/Gyro (proposée)	X,Y Final	Exp.	Carrée	? (plan)	Position fin	?
	EKF						
	UKF						
[242]	Flot optique (proposée)	Temps de calcul	Exp.	Complexe	Extérieur	GPS	?
	SIFT + RANSAC	Distance X,Y, $\theta$					

TABLE 3.1 – Localisation locale : Synthèse de l'analyse bibliographique

Il existe un très large corpus scientifique autour des méthodes de localisation locales dont sont issues les travaux de la Table 3.1. Les approches locales permettent d'analyser "simplement" l'environnement expérimental à l'aide de capteurs extéroceptifs en détectant des amers caractéristiques pour localiser le robot. Les amers employés dépendent des capteurs extéroceptifs déployés en privilégiant les marqueurs visuels naturels [19] ou artificiels [21] lorsque l'on utilise des caméras embarquées. De façon plus originale [60] fait appel à une mesure du champ

magnétique induit par des aimants placés dans l'environnement pour recalibrer la dérive odométrique. Cela conduit à la proposition d'une grande variété de méthodes dans de nombreux contextes applicatifs. Les métriques adoptées font pratiquement toutes appel à l'évaluation de la distance entre la position estimée et une réalité terrain. L'obtention de la réalité terrain privilégie le plus souvent des mesures faites avec le robot à l'arrêt (point de passage). Par exemple dans [78] le robot stoppe régulièrement sa progression pour permettre de mesurer manuellement sa position réelle. [20] ne s'intéresse qu'à la position finale du robot. La réalité terrain est plus rarement obtenue à l'aide de dispositifs de localisation externes (laser, GPS) qui sont aussi employés pour déterminer la position réelle du robot. [214] fait appel à un laser extérieur détectant une balise embarquée, et exploite les performances constructeur attendues de ce dernier. [242] utilise un GPS en extérieur pour évaluer la réalité terrain. La plupart des auteurs font appel à des trajectoires simples (droites ou carrées) [214] [20] [78]. Plus rarement en extérieur on rencontre des trajectoires plus complexes en forme de "huit" comme dans [19] ou de "U" comme dans [242], dictées par les voies de roulement empruntées. Enfin, malheureusement, le contrôle mis en œuvre pour déterminer les mesures est très rarement évoqué sauf dans [214] qui annonce clairement que le robot est téléopéré. En ce qui concerne les précisions affichées la plupart n'évoquent qu'une distance à la réalité terrain mixant ainsi l'erreur en X et Y. Les distances à la réalité terrain affichées par les méthodes de localisation dans les travaux de la Table 3.1 varient tout au long des expérimentations effectuées. Dans les courbes présentées ces distances s'inscrivent dans un intervalle compris entre 10 cm et plus d' 1 m. En moyenne les erreurs annoncées vont d'une dizaine de cm à une cinquantaine. Ce ne sont que des moyennes et aucune garantie n'est donc affichée. Les travaux différenciant X et Y montrent une disparité importante entre ces deux dimensions comme pour [242] et dans une moindre mesure [20]. Le cap est simplement étudié dans [242] qui présente une incertitude de 5°.

Source	Méthodes	Perf.	Test	Traj.	Env.	R.T.	Ctrl
[10]	CNN (proposée)	Confiance	Simu.	Carrée	Intersection	?	?
	AMCL	Distance $\theta$	Exp.				
[146]	PTPMS (proposée)	X,Y, $\theta$ Temps	Exp.	?	Couloirs	AT960 et T-MAC	?
	MCL						
[244]	PFOI (proposée)	Distance	Simu. DataSet	Carrée	Couloirs	MCL 1000 particules	?
	PFI						
[102]	Grille floue	Distance $\theta$	Exp.	Complexe	Bureau et couloirs	points de passage	Téléop.
[233]	SA-MCL (proposée)	Sigma X,Y	Exp.	Complexe	Couloirs	Amers externes	?
	AMCL						

TABLE 3.2 – Localisation globale : Synthèse de l'analyse bibliographique

On trouve dans la littérature moins de travaux faisant référence aux méthodes de localisation globales car elles sont plus complexes à déployer et à évaluer. Il faut en effet être capable de fournir au robot une carte complète de l'environnement ce qui n'est pas toujours facile. Toutes les méthodes de la Table 3.2 sont déployées en intérieur. Les approches de Monte-Carlo sont souvent employées comme point de comparaison avec la méthode étudiée comme dans [10], [146] ou [233]. Sa relative simplicité et sa robustesse expliquent sans doute sa popularité. Comme pour les méthodes locales, la distance séparant la position réelle du robot de celle estimée est souvent utilisée pour évaluer la performance des approches globales comme pour [10], [244] et [102]. Les trajectoires considérées peuvent être simples ([244] [10]), ou complexes ([102], [233]).

Parfois leur nature n'est même pas mentionnée [146]. On peut remarquer que l'incertitude en  $\theta$  est aussi plus souvent étudiée [10], [146] [102]. Là encore le contrôle du robot est exceptionnellement mentionné [102]. Les résultats sont obtenus à partir de relevés pré-enregistrés puis comparés à la réalité terrain lorsque cela est possible. Une grande variété d'approches est mise en œuvre pour l'obtenir. [233] emploie des amers artificiels référencés mais non utilisés par la méthode étudiée. [102] fait un post-traitement des relevés lasers pour trianguler la position absolue du robot. Parfois même, certains auteurs comme [244] font appel à une autre approche globale, comme un filtre particulaire sur-dimensionné (1000 particules), pour obtenir une référencement terrain. Enfin, certains comme [10] ne précisent même pas comment leur vérité terrain est obtenue. Les incertitudes affichées par les méthodes globales sont équivalentes à celles obtenues à l'aide des méthodes de localisation locales. Elles sont en général de l'ordre quelques dizaines de centimètres en revanche la disparité entre X et Y est moins prononcée. Les incertitudes de cap restent comparables à celles obtenues avec des approches locales.

Source	Méthodes	Perf.	Test	Traj.	Env.	R.T.	Ctrl
[63]	Set Inversion Intervals	Incertainude Temps de calcul	Exp.	Statique	Intérieur	Placement robot	–
[99]	IAL MCL	Temps de Calcul	Exp.	Rectiligne	Couloir	?	?

TABLE 3.3 – Méthodes de localisation par intervalle : Synthèse de l'analyse bibliographique

Les méthodes de localisation présentées dans le tableau 3.3 référencent des approches par intervalles moins répandues que les précédentes. Peu de travaux y font appel sans doute puisque ces démarches tendent à maximiser l'incertitude tout en apportant une garantie. L'absence de comparatif immédiat pousse généralement à explorer d'autres métriques plus proches de la performance de la méthodologie présentée que véritablement de la performance de localisation. Par exemple [63] s'intéresse au temps de calcul de l'incertitude de localisation (algorithme SIVIA) en fonction des dimensions du pavé de localisation correspondant d'un robot évoluant dans une salle vide de (9.4m \* 6.4m) instrumentée. Le cap n'est donc pas étudié. Le robot équipé d'une caméra panoramique est placé statiquement à différentes poses dans l'environnement. Les valeurs des pavés mesurés sont comparées à une réalité terrain obtenue à partir de caméras fixes placées dans l'environnement. Plus on veut être précis, plus le temps de calcul augmente évidemment et ne reste pas toujours compatible avec une utilisation en ligne lorsque la précision s'accroît (tps > 1 mn pour précision millimétrique).

Dans [99], les auteurs utilisent un lidar 360° dans un environnement intérieur instrumenté et analysent statiquement, ou en suivant des trajectoires rectilignes, les dimensions des pavés d'incertitudes. Les valeurs affichées vont de quelques centimètres dans un environnement de (10m \* 3m) à près d'une demi-mètre dans couloir de (26m \* 4m) avec des incertitudes de cap voisines de 3°. La réalité terrain est obtenue en faisant appel à un filtre particulaire et l'erreur est analysée hors-ligne. La durée de calcul est de l'ordre de 1s.

Pour ces travaux, les environnements restent donc de faibles dimensions, et les durées d'obtention des pavés d'incertitudes restent en général trop importantes pour une utilisation permanente en temps réel.

### 3.5.4 Incertitudes affichées : Enseignements

Plus globalement l'analyse qui vient d'être effectuée démontre, s'il en était encore besoin, que la problématique de la localisation est évidemment centrale en robotique puisque la littérature

regorge d'un nombre très important de contributions au sein desquelles nous espérons avoir sélectionné un échantillon représentatif d'articles.

On peut avancer que les méthodes de localisation les plus largement représentées, et de loin, dans les travaux de recherche, sont les méthodes locales avec en premier chef l'utilisation de l'algorithme de fusion le plus répandu, à savoir le filtre de Kalman et tous ses dérivés.

Pour qualifier les performances d'une méthode de localisation donnée, toutes ces études suivent logiquement, mais plus ou moins rigoureusement, le même processus d'analyse. Après avoir présenté les fondamentaux théoriques de l'approche proposée, celle-ci est déployée dans un environnement donné en simulation ou expérimentalement, les informations de localisation recueillies sont alors comparées à une réalité terrain et/ou à d'autres méthodes afin d'en estimer les performances.

Intéressons nous donc maintenant aux incomplétudes que nous avons identifiées dans les articles de notre étude.

- Les conditions expérimentales sont toujours présentées. Les trajectoires peuvent être très variées mais certaines sont particulièrement récurrentes comme les boucles (cercle, carré) et les droites (65% des articles analysés). On peut cependant s'étonner des dimensions particulièrement réduites de certaines zones d'expérimentation et du nombre réduit d'expériences réalisées (40% < 10m). Il faut aussi souligner qu'évidemment une méthode de localisation est fortement dépendante de l'environnement choisi (fort aliasing perceptuel ou non par exemple) mais que l'adaptabilité d'une méthode n'est, à notre connaissance, que rarement voire jamais évoquée. Enfin les conditions expérimentales peuvent influencer significativement les performances des capteurs employés. Là encore celles-ci sont rarement clairement présentées ce qui laisse supposer que les expérimentations ont été effectuées dans des conditions expérimentales optimales.
- La nature du schéma de contrôle adopté pour effectuer une expérimentation est importante pour pouvoir estimer les performances de localisation d'une méthode. En boucle ouverte ou en téléopération on va suivre une trajectoire donnée sans asservir la trajectoire du système sur les informations de localisation issues de la méthode étudiée. En boucle fermée celles-ci asserviront le suivi de chemin. Cette indication n'est que très rarement mentionnée dans les articles (31%). Cela pose donc la question de la méthode de contrôle employée pour obtenir les mesures permettant d'estimer la performance de localisation. Lorsque cela est mentionné les auteurs travaillent en boucle ouverte ce qui facilite la comparaison des méthodes étudiées.
- Le choix / la configuration des méthodes de localisation employées à titre de comparaison porte parfois à discussion. En effet si l'on utilise une méthode ayant des performances limitées on pourra sans nul doute conclure que la méthode exposée est plus performante. C'est par exemple souvent le cas lorsque l'odométrie est employée comme élément de comparaison pour évaluer la performance de localisation d'une méthode. Ce choix reste cependant discutable en raison de la dérive odométrique mais reste "valide" sur de faibles distances.
- La caractérisation des incertitudes de la réalité terrain est un des points les plus sensibles pour établir rigoureusement les performances de pose d'une méthode de localisation. En l'absence de système externe de type *ViCon* impossible à déployer sur un environnement de grandes dimensions, la qualification des incertitudes de la réalité terrain (et par la même de la méthode de localisation étudiée) nécessite un travail long et fastidieux. Cet aspect pourtant central pour déterminer la précision d'une méthode est très rarement abordé dans les articles (34 %). Aucun article, dans notre sélection, ne caractérise expérimentalement les performances du système employé pour obtenir la réalité terrain ou,

au mieux, se contente d'utiliser les informations constructeur. En son absence, on peut, à juste titre, s'interroger sur la validité des performances affichées.

- On peut aussi souligner qu'aucun article ne réalise une analyse, même limitée, de l'impact des différentes sources d'incertitudes sur les données de localisation. Entre autre les incertitudes des montages mécaniques, surtout en rotation peuvent avoir un impact non négligeable sur les performances obtenues.
- Les performances des méthodes de localisation recouvrent d'autres paramètres (temps de convergence, de calcul, etc..) que le triplet de pose. Cependant si l'on se restreint à ce dernier on peut identifier les limitations suivantes :
  - La performance de localisation  $(X,Y)$  est toujours étudiée mais elle est parfois présentée de façon indirecte en calculant une distance entre la réalité terrain adoptée et les estimations faites à l'aide de la méthode étudiée (76% des articles). Ce choix brouille un peu l'analyse en fusionnant ces deux dimensions au risque de passer sous silence une sensibilité particulière selon un axe donné.
  - Si la performance de localisation est toujours étudiée, celle liée au cap  $(\theta)$  n'est présentée que dans 23% des articles analysés. Il est clair qu'il est plus difficile de qualifier des erreurs d'orientation. Cette omission reste cependant regrettable lors de l'utilisation, en boucle fermée, d'une méthode de localisation, car cela va impacter la qualité du suivi de trajectoire mis en place en induisant des oscillations plus ou moins importantes.

De façon à chercher à répondre aux incomplétudes qui viennent d'être identifiées, l'étude de la performance de localisation des méthodes déployées dans le cadre de ce manuscrit suivra les principes suivants :

- Nous avons étudié la performance de localisation de notre robot le long de, et perpendiculairement à, l'abscisse curviligne de la trajectoire suivie par le robot c'est à dire selon  $X_s$  et  $Y_s$ . Ce choix nous a semblé judicieux afin de ne pas brouiller l'analyse de la localisation puisque les couloirs traversés ne sont pas toujours parallèles aux mêmes axes du repère Monde.
- Nous étudions la performance de pose du robot. Nos travaux incluent donc l'analyse de l'incertitude en  $\theta$ .
- Conditions expérimentales : Les modèles prédictifs de performance de localisation ont été établis au sein d'une zone expérimentale d'une trentaine de mètres de long avec un éclairage optimal. Des trajectoires rectilignes ont été utilisées et un nombre significatif d'expérimentations a été déployé.
- Nous avons spécifié le type de schéma de contrôle employé pour qualifier nos méthodes (boucle ouverte ou fermée).
- Nous avons apporté une attention particulière à la caractérisation de notre réalité terrain.

Pour terminer nous allons préciser le lien qui existe pour nous entre la garantie de localisation et l'incertitude des méthodes utilisées.

### 3.5.5 Méthode, incertitude et garantie de Localisation

Les travaux exposés dans l'analyse précédente font nécessairement appel à un observateur indépendant (et souvent externe) pour qualifier les performances des méthodes de localisation présentées. La réalité terrain ainsi obtenue doit être de qualité pour pouvoir évaluer le plus finement possible les erreurs et incertitudes des méthodes étudiées.



La méthodologie PANORAMA a besoin d'un observateur indépendant et embarqué des performances que l'on cherche à garantir de façon à pouvoir suivre, tout au long d'une mission, l'évolution des marges de performance. Cela est facilement accessible pour les dimensions d'Énergie (wattmètre) et de Durée (horloge). Pour l'axe Localisation il nous faudrait donc disposer d'une méthode de référence pour pouvoir estimer la performance de la méthode de localisation employée à un instant donné. Dès lors cela revient à dire que nous ne pourrions faire appel à cette méthode de référence, pourtant disponible et utilisée, pour exécuter notre mission, puisque nous ne pouvons la comparer à elle-même. On ne peut donc pas observer et donc garantir une erreur de localisation.

Pour pallier cette difficulté, nous avons donc choisi de nous appuyer sur le modèle d'incertitude de nos méthodes de localisation. Pour cela nous allons les construire, à l'aide d'une réalité terrain, à partir des modèles d'erreur que nous allons établir expérimentalement. Une fois embarqués, et donc en l'absence d'une réalité terrain objective, ces modèles d'erreur deviendront donc des modèles d'incertitude que nous utiliserons pour suivre l'évolution de la performance de Localisation tout au long de la mission. Ils seront joués entre deux mesures.

Il faut cependant souligner que si ces modèles d'incertitude ont été correctement construits et que les méthodes associées sont bien utilisées dans les zones de l'environnement où elles sont opérationnelles, alors l'incertitude prédite se projette en garantie de Localisation.

## 3.6 Conclusion

Ce chapitre nous a permis de présenter l'ensemble des éléments participant ou impactant le processus de détermination de la pose d'un robot mobile. Cela nous a conduit à évoquer, les différents référentiels impliqués, le robot lui-même au travers de ses vues matérielle et logique. Puis, après avoir défini les deux problématiques centrales de localisation que sont le suivi de posture et la localisation globale nous avons réalisé un tour d'horizon des différentes approches permettant d'y répondre tout en évoquant leurs principales limitations. Aucune d'elles ne propose de modèle prédictif. Notre analyse s'est alors focalisée sur la notion d'incertitude, qui vient de fait brouiller la capacité du robot à connaître sa réelle posture. Nous avons alors analysé de façon critique les précisions affichées par un ensemble de travaux représentatifs des approches de localisation utilisées en robotique. Cette étude a permis de conclure que les méthodes de suivi de posture étaient les plus largement étudiées. D'un point de vue quantitatif les travaux peuvent afficher des erreurs **moyennes** de quelques dizaines de centimètres mais sans apporter de garantie et en ne commentant pas les extrema rencontrés. Cependant l'analyse réalisée a souligné que nombre de ces travaux souffraient d'incomplétudes tant au niveau de l'identification des incertitudes impactant les méthodes, des descriptions des conditions expérimentales, mais surtout de la qualification de la réalité terrain de référence. Enfin nous avons explicité la raison pour laquelle nous nous focaliserons dans la suite de ce document sur l'incertitude de nos méthodes pour chercher à garantir la Localisation.

Ce chapitre constitue la grille de description et d'analyse que nous allons suivre, tout au long des deux prochaines parties de ce document. Dans un premier temps nous présenterons et détaillerons la mission robotique qui va sous-tendre notre discours ainsi que l'organisation matérielle et logique du robot qui devra l'effectuer. Puis, dans un second temps, nous chercherons à proposer des modèles prédictifs pour les approches de localisation qui seront déployées pour cette mission. Et cela en cherchant à caractériser les principales incertitudes qui viendront entacher le processus de localisation.

# Chapitre 4

## CONTEXTE EXPÉRIMENTAL ET INCERTITUDES

Afin de déployer notre approche d'estimation des incertitudes de localisation, nous allons nous appuyer dans ce chapitre sur la description de la mission cible et de la plateforme robotique qui sera employée. Tout au long de notre discours nous tenterons de pointer les principales incertitudes qui viennent affecter la précision de localisation/pose et que nous pouvons estimer. Le chapitre suivant nous permettra de les combiner pour chercher à établir les modèles prédictifs d'incertitude de localisation qui nous sont nécessaires pour la méthode PANORAMA.

En préambule, précisons ici que la valeur des incertitudes que nous afficherons sera inévitablement entachée par les modestes moyens que nous avons eu à notre disposition pour compléter l'instrumentation de notre plateforme robotique et pour cartographier et instrumenter l'environnement d'expérimentation. D'une part les constructions mécaniques ajoutées sur le robot pour l'équiper de nouveaux capteurs ne sont pas de qualité industrielle. D'autre part, notre cartographie de l'environnement repose sur l'utilisation d'un décimètre à ruban, d'un mètre à ruban et d'un télémètre laser.

Nous commencerons par présenter une vue d'ensemble du calcul d'incertitudes qui viendra affecter les méthodes de localisation cela tant pour les approches locales, que globales. Dans un second temps nous présenterons la mission de patrouille en milieu intérieur que nous chercherons à exécuter, à terme, en garantissant, ne l'oublions pas, des objectifs de performance relevant de la sécurité, l'énergie, la durée et la localisation. Cela nous conduira bien évidemment à nous intéresser à la représentation de l'environnement dans lequel se déplacera le robot. Enfin, au travers de la description de la vue matérielle et logicielle de la plateforme robotique employée, nous viendrons, au fur et à mesure, identifier et caractériser les principales incertitudes qui impacteront l'estimation finale de localisation.

### 4.1 Localisation, pose et incertitudes : Vue d'ensemble

Cette partie va nous permettre de présenter la formulation générale du calcul de la pose d'un robot. Nous considérerons deux cas de figure : une approche locale de suivi de posture, et une approche globale faisant appel à un partitionnement de l'environnement de type grille. Puisque nous nous inscrivons dans une logique de garantie, nous allons donc faire appel à une démarche ensembliste par intervalles pour déterminer la taille des pavés pouvant être associés à la pose du robot.

Il est important de préciser à quel moment l'incertitude de localisation est considérée dans notre étude. Pour déterminer la commande à appliquer, le guidage doit connaître la localisation courante du robot. Cette dernière est obtenue à partir de mesures capteurs antérieures. Le système robotique s'est donc déplacé depuis l'obtention de l'information de localisation. Il faudrait donc inclure dans l'incertitude de localisation une estimation du déplacement effectué depuis l'obtention des mesures capteurs. Cependant, comme nous le verrons à la fin de ce chapitre, cette incertitude d'origine logicielle, peut être en partie compensée lors du calcul de la commande.

Nous considérerons donc dans notre analyse que l'incertitude de localisation est celle observée à la date d'obtention des mesures capteurs.

Enfin rappelons que nous nous intéressons à la pose du robot en 2 dimensions dans son environnement. Nous serons amenés à considérer les référentiels suivants dans nos calculs :

- Le référentiel *Monde*  $\mathcal{R}_M$  est un référentiel fixe, défini dans la carte, au sein duquel la pose du robot doit être déterminée.
- Le référentiel *Robot*  $\mathcal{R}_R(t)$  est un référentiel mobile dans  $\mathcal{R}_M$ .
- Le référentiel *Capteur*  $\mathcal{R}_C(t)$  est un référentiel qui est attaché à chacun des capteurs extéroceptifs embarqués du robot. Il est donc mobile dans  $\mathcal{R}_M$  et immobile dans le référentiel robot  $\mathcal{R}_R$  avec lequel il présente une matrice de transformation constante.
- Le référentiel *Marqueur* que nous noterons  $\mathcal{R}_{QR}$  est un référentiel 2D ou 3D fixe dans  $\mathcal{R}_M$  qui est attaché aux marqueurs placés dans l'environnement.
- Par ailleurs dans la suite nous travaillons avec des vecteurs en coordonnées homogènes

Nous allons maintenant présenter les équations générales employées pour déterminer la posture du robot dans le cas d'une approche locale de localisation par suivi de posture, et dans celui d'une approche globale localisant de façon absolue le robot au sein d'une grille.

#### 4.1.1 Localisation et pose : Approche locale de suivi de posture

Lorsque l'on effectue un suivi de posture on vient corriger la dérive des informations odométriques proprioceptives dès que possible, et lorsque cela est nécessaire, en faisant appel à des informations extéroceptives permettant de relocaliser globalement le robot.

On peut dans un premier temps exprimer de façon très générale l'évolution de la localisation d'un robot entre  $(t - \Delta t)$  et  $t$  dans  $\mathcal{R}_M$  par l'Équation 4.1 qui reste évidemment valide dans le cas d'une navigation à l'estime :

$$\begin{bmatrix} O_{\mathcal{R}_R}(t)|_{\mathcal{R}_M} \end{bmatrix} = MT_{\mathcal{R}_M|\mathcal{R}_R(t-\Delta t)} \bullet \begin{bmatrix} O_{\mathcal{R}_R}(t)|_{\mathcal{R}_R(t-\Delta t)} \end{bmatrix} \quad (4.1)$$

$\begin{bmatrix} O_{\mathcal{R}_R}(t)|_{\mathcal{R}_M} \end{bmatrix}$  représente les coordonnées de l'origine de  $\mathcal{R}_R$  dans  $\mathcal{R}_M$  à l'instant  $t$ .

$\begin{bmatrix} O_{\mathcal{R}_R}(t)|_{\mathcal{R}_R(t-\Delta t)} \end{bmatrix}$  représente les coordonnées de l'origine de  $\mathcal{R}_R$  à l'instant  $t$  exprimées dans  $\mathcal{R}_R$  à l'instant  $(t - \Delta t)$ .

$MT_{\mathcal{R}_M|\mathcal{R}_R(t-\Delta t)}$  représente la matrice de transformation reliant  $\mathcal{R}_R$  à l'instant  $(t - \Delta t)$  à  $\mathcal{R}_M$ .

En utilisant itérativement cette équation il est donc possible d'inférer l'évolution du robot dans  $\mathcal{R}_M$ .

Considérons maintenant que le robot recale sa position grâce à une mesure extéroceptive permettant d'identifier un amer géolocalisé dans l'environnement. Pour cette mesure proprement dite on peut établir l'Équation 4.2 qui exprime la position du repère Marqueur  $\mathcal{R}_{QR}$  par rapport au repère capteur  $\mathcal{R}_C(t)$  à l'instant t :

$$\begin{bmatrix} O_{\mathcal{R}_{QR}|\mathcal{R}_C(t)} \end{bmatrix} = MT_{\mathcal{R}_C(t)|\mathcal{R}_{QR}} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (4.2)$$

$MT_{\mathcal{R}_C(t)|\mathcal{R}_{QR}}$  représente la matrice de transformation de  $\mathcal{R}_{QR}$  à  $\mathcal{R}_C$  à l'instant t.

Donc si l'on veut relier  $\mathcal{R}_R(t)$  à  $\mathcal{R}_{QR}$  on peut écrire (Équation 4.3) :

$$\begin{bmatrix} O_{\mathcal{R}_{QR}|\mathcal{R}_R(t)} \end{bmatrix} = MT_{\mathcal{R}_R|\mathcal{R}_C} \bullet MT_{\mathcal{R}_C(t)|\mathcal{R}_{QR}} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (4.3)$$

$MT_{\mathcal{R}_R|\mathcal{R}_C}$  représente la matrice de transformation de  $\mathcal{R}_C$  à  $\mathcal{R}_R$ .

Mais l'on cherche la localisation du robot à l'instant t  $\begin{bmatrix} O_{\mathcal{R}_R(t)} \end{bmatrix}$  dans le référentiel Monde  $\mathcal{R}_M$ . l'Équation 4.4 peut donc être écrite :

$$\begin{bmatrix} O_{\mathcal{R}_R(t)|\mathcal{R}_M} \end{bmatrix} = MT_{\mathcal{R}_M|\mathcal{R}_{QR}} \bullet MT_{\mathcal{R}_{QR}|\mathcal{R}_R(t)} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (4.4)$$

Que l'on peut aussi écrire sous la forme de l'Équation 4.5 :

$$\begin{bmatrix} O_{\mathcal{R}_R(t)|\mathcal{R}_M} \end{bmatrix} = MT_{\mathcal{R}_M|\mathcal{R}_{QR}} \bullet MT_{\mathcal{R}_R(t)|\mathcal{R}_{QR}}^{-1} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (4.5)$$

Où  $MT_{\mathcal{R}_R(t)|\mathcal{R}_{QR}}^{-1}$  représente la matrice inverse de transformation de  $\mathcal{R}_{QR}$  à  $\mathcal{R}_R(t)$  à l'instant t.

$$\text{Avec } MT_{\mathcal{R}_R(t)|\mathcal{R}_{QR}}^{-1} = MT_{\mathcal{R}_C(t)|\mathcal{R}_{QR}}^{-1} \bullet MT_{\mathcal{R}_R|\mathcal{R}_C}^{-1}$$

$$\text{Mais on peut aussi écrire } MT_{\mathcal{R}_R(t)|\mathcal{R}_{QR}}^{-1} = MT_{\mathcal{R}_{QR}|\mathcal{R}_C(t)} \bullet MT_{\mathcal{R}_C|\mathcal{R}_R} = MT_{\mathcal{R}_{QR}|\mathcal{R}_R(t)}$$

En ce qui concerne l'orientation, elle est obtenue simplement en sommant les orientations des différents repères utilisés (Équation 4.6) :

$$\theta_{\mathcal{R}_M|\mathcal{R}_R(t)} = \theta_{\mathcal{R}_M|\mathcal{R}_{QR}} + \theta_{\mathcal{R}_{QR}|\mathcal{R}_C(t)} + \theta_{\mathcal{R}_C|\mathcal{R}_R} \quad (4.6)$$

Où  $\theta_{\mathcal{R}_1|\mathcal{R}_2}$  représente l'angle du référentiel  $\mathcal{R}_2$  par rapport au référentiel  $\mathcal{R}_1$ .

### 4.1.2 Localisation et pose : Approche globale orientée grille

Nous considérerons maintenant une méthode globale où les informations extéroceptives sont utilisées directement par une méthode de localisation, faisant appel à une grille 2D, sans connaissance préalable de la posture précédente du robot.

Dans ce cas de figure la méthode de localisation on va dans un premier temps pouvoir écrire (Équation 4.7) :

$$\boxed{\left[ O_{\mathcal{R}_R}(t) \right]_{\mathcal{R}_M} = MT_{\mathcal{R}_R|\mathcal{R}_C}^{-1} \bullet \left[ O_{\mathcal{R}_C}(t) \right]_{\mathcal{R}_M} \quad (4.7)}$$

Où  $\left[ O_{\mathcal{R}_C}(t) \right]_{\mathcal{R}_M}$  représente la position du référentiel Capteur  $\mathcal{R}_C$  dans le référentiel Monde  $\mathcal{R}_M$  à l'instant  $t$ .

Puisque nous disposons déjà d'une localisation du référentiel Capteur dans le référentiel Monde, il suffit maintenant de ramener le référentiel Robot dans le référentiel Capteur ce qui est fait par l'entremise de la matrice de transformation inverse  $MT_{\mathcal{R}_R|\mathcal{R}_C}^{-1}$

De la même façon que pour la localisation locale, pour obtenir la pose du robot, on peut écrire pour les rotations (Équation 4.8) :

$$\boxed{\theta_{\mathcal{R}_M|\mathcal{R}_R}(t) = \theta_{\mathcal{R}_C|\mathcal{R}_R} + \theta_{\mathcal{R}_M|\mathcal{R}_C}(t) \quad (4.8)}$$

Les formulations générales de la localisation et de la pose du robot étant explicitées, intéressons nous maintenant à la prise en compte des incertitudes.

### 4.1.3 Localisation, pose et incertitudes

Dans les équations que nous venons d'énoncer pour les méthodes locales et globales, les données des équations sont entachées d'erreurs qui se distribuent au travers des différents termes qui les composent :

- $MT_{\mathcal{R}_R(t)|\mathcal{R}_{QR}}^{-1}$  intègre d'une part les incertitudes de mesures capteurs (calibration, précision, etc.) et les incertitudes logicielles de calcul (arrondi), et d'autre part les incertitudes mécaniques de positionnement du repère capteur.
- $MT_{\mathcal{R}_M|\mathcal{R}_{QR}}$  intègre les incertitudes de connaissance que nous avons sur la localisation des marqueurs.
- $\left[ O_{\mathcal{R}_C}(t) \right]_{\mathcal{R}_M}$  intègre les incertitudes liées à la taille de la grille choisie dans la formulation de la méthode globale.
- $\theta_{\mathcal{R}_1|\mathcal{R}_2}$  intègre les incertitudes de rotation liées au passage d'un repère à un autre.

Une façon naturelle, efficace et garantie de prendre en compte ces incertitudes va être de faire appel au calcul ensembliste. Les incertitudes sur les mesures de translation ou de rotation seront donc représentées par des pavés, non nécessairement centrés, caractérisés par une valeur minimale, et une valeur maximale.

Le calcul ensembliste permet de prendre en compte les fonctions trigonométriques présentes dans les matrices de rotation qui sont impliquées dans le calcul matriciel menant à l'expression finale de la pose garantie que nous recherchons :

$$\left[ PO_R(t) \right] = \left[ [x_R(t)], [y_R(t)], [\theta_R(t)] \right]^T$$

De façon générale l'incertitude de localisation peut se décomposer en 2 parties. L'une où l'on se recalcule globalement de façon discrète. L'autre correspond à l'évolution de l'incertitude entre 2 instants de recalage. Ainsi donc, plus la durée en 2 recalages va être importante, plus l'incertitude de localisation à un instant  $t$  va croître à partir du dernier instant de recalage  $t_r$ . On peut alors déduire le modèle d'incertitude général 4.9 dans lequel :

- $\left[ O_{\mathcal{R}_R}(t)|_{\mathcal{R}_M} \right]$  est l'intervalle d'incertitude de position du robot à l'instant  $t$  dans le repère Monde.
- $\left[ MT_{\mathcal{R}_M|\mathcal{R}_R(t_r)} \right]$  est la matrice de rotation homogène constituée à partir des intervalles d'incertitudes du recalage obtenu  $t_r$  antérieur à  $t$ .
- $\left[ O_{\mathcal{R}_R}(t)|_{\mathcal{R}_R(t_r)} \right]$  est le vecteur de déplacement estimé du robot entre  $t_r$  et  $t$ . Il correspond aux intervalles d'incertitudes que nous avons sur le déplacement du robot sur cette période.

$$\left[ O_{\mathcal{R}_R}(t)|_{\mathcal{R}_M} \right] = \left[ MT_{\mathcal{R}_M|\mathcal{R}_R(t_r)} \right] \bullet \left[ O_{\mathcal{R}_R}(t)|_{\mathcal{R}_R(t_r)} \right] \quad (4.9)$$

L'équation 4.9 ne nous renvoie qu'un pavé suivant  $x$  et  $y$ . Pour obtenir l'incertitude de pose  $\left[ PO_R(t) \right]$  il faut aussi considérer l'incertitude d'orientation  $\theta$  (Equation 4.8).

Dans la suite nous allons donc chercher à caractériser les pavés d'incertitudes impliqués dans chacune des étapes de construction de la localisation du robot à partir des mesures capteurs. Le pavé final sera obtenu en utilisant les règles du calcul ensembliste à partir des équations définies et des incertitudes observées. Ce calcul aurait pu faire appel à un outil de calcul ensembliste [117] [55], mais, pour des raisons de simplicité, puisque nous avons choisi de ne pas utiliser de contracteurs, nous avons préféré développer nos propres fonctions dans l'outil Matlab<sup>®</sup>, en s'étant au préalable assuré de la justesse des calculs effectués.

Dans les sections suivantes nous allons définir, sur notre plateforme, les différentes matrices de transformation impliquées dans le calcul de la pose du robot ainsi que les pavés d'incertitudes qui y sont attachés.

## 4.2 Contexte expérimental : Mission de patrouille

Cette partie décrit tout d'abord la mission de patrouille qui doit être effectuée par notre robot mobile, puis précise la représentation métrique qui est associée à l'environnement d'évolution.

## 4.2.1 Généralités

La **mission de patrouille**, *Patrouiller* qui nous intéresse vise à inspecter l'état (ouvert ou fermé) de deux vannes placées dans les sous-sols du laboratoire à l'aide d'un robot mobile totalement autonome revenant à son point de départ.

La localisation en environnement intérieur est une tâche difficile à plus d'un titre. D'une part, comme nous l'avons déjà souligné, les constructions humaines, et particulièrement les bâtiments, sont composés d'un ensemble de pièces reliées par des couloirs rectilignes. Ces espaces, qui sont donc fortement structurés et symétriques, favorisent l'émergence du phénomène d'aliasing perceptuel. D'autre part, nous ne disposons pas, comme en extérieur avec le GPS, d'un système omniscient, capable de localiser presque en tout instant (problème de perte de constellation), dans le référentiel Monde, et avec une précision limitée mais connue, un système mobile.

La Figure 4.1 présente une vue d'ensemble de l'environnement de la mission et va nous permettre de la décrire. On distingue 3 zones. La halle d'expérimentation où se trouve initialement le robot, un premier couloir CR1 que nous supposons ouvert au public et donc où le robot peut être amené à éviter des obstacles, un second couloir CR2 où le robot est le seul à pouvoir circuler. On peut remarquer que cet environnement intègre aussi des zones où des vitres sont présentes réellement ou de façon simulée (voisinage Vanne 1). Enfin on peut constater que des amers artificiels géolocalisés ont été ajoutés tout au long de la mission, pour faciliter le recalage spatial du robot, et/ou pour localiser les vannes.

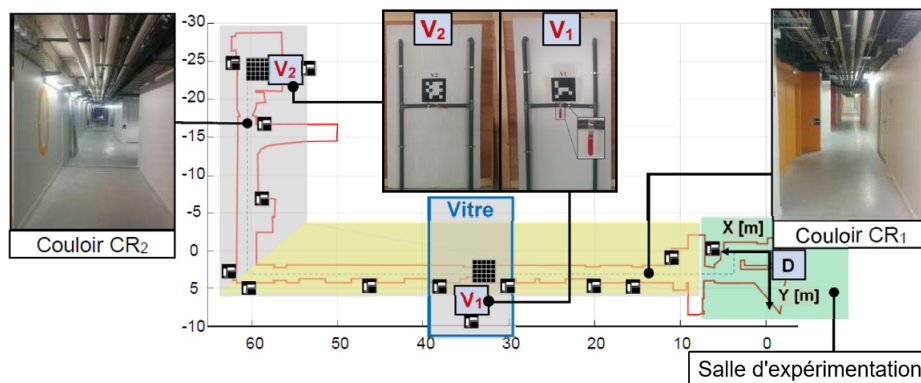


FIGURE 4.1 – Vue d'ensemble de l'environnement de la mission

Ainsi la mission *Patrouiller* peut se décomposer en une séquence d'objectifs (Table 4.1) faisant appel à une ou plusieurs tâches robotiques.

Objectif	Description	Tâches robotiques
O1	Aller à V1	DEP / LOC
O2	S'orienter vers V1	ROT / LOC
O3	Inspecter l'état de V1	ANIM
O4	S'orienter vers le chemin	ROT / LOC
O5	Aller à V2	DEP / LOC
O6	S'orienter vers V2	ROT / LOC
O7	Inspecter l'état de V2	ANIM
O8	Se retourner	ROT / LOC
O9	Aller à D	DEP / LOC

DEP : DEPlacement – LOC : LOCALisation – ROT : ROTation – ANIM : ANalyse IMage

TABLE 4.1 – Objectifs de la mission *Patrouiller*

On constate évidemment que la tâche de localisation est présente dans toutes les phases de la mission où le robot est en mouvement, mais aussi lors de l'analyse de l'état de la vanne puisque le robot doit être correctement positionné vis-à-vis de cette dernière.

Pour exécuter sa mission, le robot dispose d'une part d'un chemin de référence préétabli rectiligne, positionné au centre de l'espace de circulation, mais aussi d'une carte connue de l'environnement que nous allons maintenant détailler.

## 4.3 Carte de l'environnement

### 4.3.1 Vue d'ensemble

Établir une carte pour définir une représentation de l'environnement d'évolution du robot dans le référentiel Monde n'est pas une tâche aussi simple qu'il n'y paraît. En effet, le premier réflexe est de faire appel à la carte d'un bâtiment. Cependant cette dernière ne peut constituer, le plus souvent, qu'une base de travail. D'une part les informations métriques disponibles peuvent s'avérer approximatives, voire totalement fausses. De plus, la carte ne constitue qu'une vue dimensionnelle de l'espace, mais il est souhaitable qu'elle soit enrichie par un ensemble d'informations pertinentes quand à l'employabilité des capteurs tant à des fins de sécurité que de localisation. On peut aussi y adjoindre toute sorte d'informations de géolocalisation permettant de repérer des amers ou dispositifs spécifiques pouvant s'avérer utiles pour l'exécution de la mission.

La carte de l'environnement présentée à la Figure 4.2 a été établie en utilisant des instruments de mesure de précision limitée. D'une part un double décimètre à ruban, et donc souple, permettant d'avoir une précision au centimètre près. D'autre part un télémètre laser affichant une précision de  $\frac{1}{10}$  de millimètre et doté d'un niveau à bulle. Compte tenu d'une manipulation manuelle même précautionneuse, nous retiendrons une incertitude d'un millimètre pour ce dernier pour prendre en compte les erreurs d'orientation. Les cotes portées sur la carte ont été établies à partir de l'origine du référentiel Monde  $\mathcal{R}_M$  située dans le couloir 2. Nous avons supposé que les murs de ce couloir étaient parfaitement rectilignes et parallèles à l'abscisse de  $\mathcal{R}_M$ , et que les murs du couloir 1 étaient parfaitement rectilignes et parallèles à l'axe des ordonnées de  $\mathcal{R}_M$  (donc perpendiculaires au couloir 2). Ces différentes cotes ont donc été identifiées de proche en proche, en effectuant des mesures de distance entre amers de l'environnement. Compte tenu des incertitudes de nos instruments de mesures et de notre méthode de reconstruction de l'environnement nous estimons que la carte a été établie avec une incertitude de 7 centimètres. Nous la supposons cependant "parfaite" dans la suite du document et elle constituera notre référentiel absolu.

- Incertitude décimètre :  $\delta_{Deca} = 1cm$
- Incertitude télémètre laser :  $\delta_{Telm} = 1mm$
- Incertitude carte :  $\delta_{Env} = 7cm$

Incertitude 1 : Incertitudes liées à la carte de l'environnement

Sur la carte de la Figure 4.2 les informations suivantes, dont on trouvera les coordonnées absolues en Annexe 1, ont été portées :

- La position de l'origine du référentiel Monde  $\mathcal{R}_M$  et son orientation.
- Le point de départ (et d'arrivée) de la mission ainsi que celle des vannes V1 et V2.



- La position des "limites du Monde" correspondant à des passages qui ne sont pas empruntés.
- La position des zones réfléchissantes qui ne sont pas détectées par nos lasers.
- La position des différents amers artificiels (QR-Codes) positionnés dans l'environnement et géo-référencés (ID :  $n^\circ$ ) dans  $\mathcal{B}_M$ . Un ensemble de symboles (voir légende de la Figure 4.2) sont utilisés pour représenter la visibilité des QR-Codes à l'aller et au retour de la mission en fonction de l'orientation des marqueurs, de celle des caméras et du sens de déplacement du robot. 3 ensembles de QR-Codes seront employés. QRC1 regroupe les amers de couleur noire et QRC2 ceux de couleur jaune, placés verticalement. Enfin QRC3 réuni les amers de couleur bleue placés au plafond. La géolocalisation des QR-Codes est précisée en Annexe C.
- La couleur des murs, orange ou grise, a aussi été précisée, car la couleur affecte la validité des informations délivrées par les lasers.
- Une zone de vitrage, ajoutée virtuellement à proximité de la Vanne 1, au sein de laquelle les capteurs laser ne pourront être employés pour détecter les murs.
- Une zone meublée, au voisinage de la Vanne 2 a aussi été délimitée. Cette zone nécessitera d'utiliser les sonars pour détecter les obstacles car les lasers de notre robot sont positionnés trop haut pour pouvoir les apercevoir.

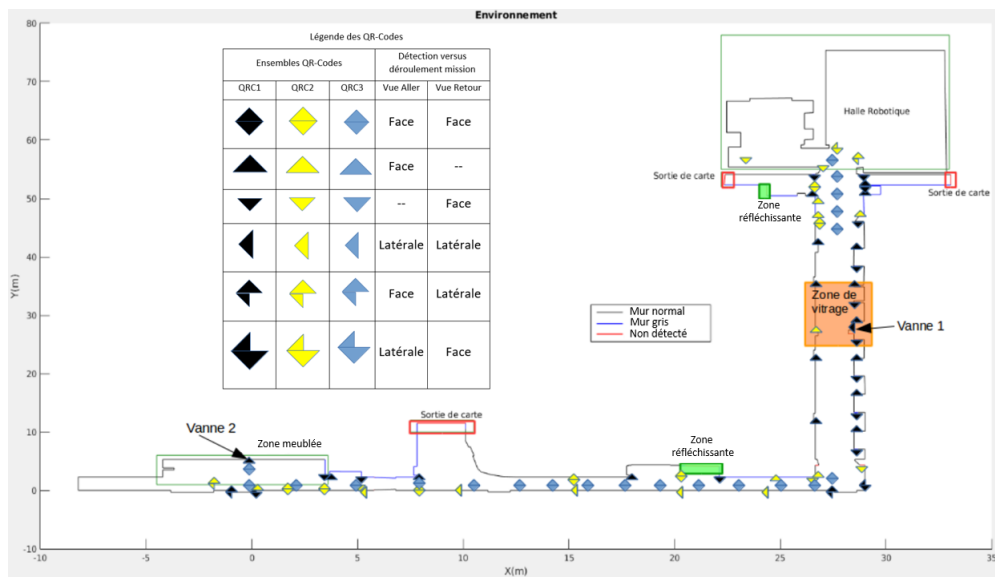


FIGURE 4.2 – Carte enrichie de l'environnement de la mission

Si nous supposons, de façon abusive, que notre carte est juste, il n'en demeure pas moins que le positionnement des amers disséminés dans l'environnement pour que le robot puisse se recalculer de temps à autre est entaché d'erreur. C'est sur ce point que nous allons maintenant nous attarder.

### 4.3.2 Instrumentation de l'environnement et incertitudes

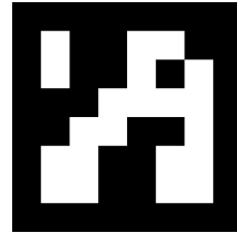
Les QR-Codes (*Quick Response Code*) sont des marqueurs visuels passifs (4.3 a)) pouvant facilement être intégrés dans l'environnement d'évolution du robot. Ils sont très aisés à produire, peu coûteux (mais il faut une caméra embarquée pour les lire), robustes aux erreurs, et permettent de stocker facilement de l'information telle qu'un référencement absolu. Ils sont assez couramment utilisés en robotique dans un environnement intérieur pour faciliter la relocalisation des systèmes robotiques [52] [239] [161] ou pour évaluer la précision d'une méthode

de localisation mise en œuvre. L'utilisation conjuguée de librairie telle que Zbar [44] et de Open-CV [176] permettent d'analyser ces QR-Codes.

Pour notre part nous avons employé la librairie ArUCo [14] [199], qui a été intégrée à OpenCV et fait appel à des capteurs visuels carrés (Figure 4.3 b)) voisins des QR-Codes. Le codage utilisé mémorise moins d'informations mais de façon plus robuste en maximisant la distance entre 2 codes voisins [88]. Le dictionnaire de correspondances n'est pas fixe, il dépend du nombre de bits de codage (usuellement 7 x 7) et du nombre d'identificateurs à générer. Une fois la caméra calibrée, la librairie Aruco, à partir de la détection des 4 coins d'un marqueur de dimension connue, renvoie les vecteurs de translation et de rotation permettant de caractériser la pose relative de la caméra par rapport au centre du marqueur.



(a) Exemple de QR-Code



(b) Marqueur Aruco

FIGURE 4.3 – Exemples de marqueurs visuels

Des marqueurs Aruco (que nous appellerons aussi QR-Codes par abus de langage), de 20 cm de côté ont été imprimés.

Ceux des ensembles QRC1 et QRC2 sont positionnés verticalement dans les couloirs 1 et 2, d'un côté ou de l'autre de ces derniers de façon à ce qu'ils se trouvent, la plupart du temps, face au robot en fonction de son sens de déplacement. Ils ont été placés, sur certains murs, ou sur des piliers saillants de l'environnement, avec précaution, en veillant à ce qu'ils soient bien horizontaux (niveau à bulle) et solidaires de leur support. En fonction du sens de déplacement du robot, un marqueur se trouvant sur un mur, face à lui à l'aller, pourra aussi être détecté au retour à  $\frac{\pi}{2}$ . Cette configuration peut entre autres se rencontrer au niveau de l'angle d'intersection entre les couloirs 1 et 2. Le code de chacun de ces marqueurs permet de connaître la pose absolue de ceux-ci dans le référentiel Monde  $\mathcal{R}_M$ .

Ceux de l'ensemble QRC3 sont positionnés au plafond à certains endroits des couloirs 1 et 2, en veillant à leur horizontalité et à leur orientation (axes X et Y aussi parallèles que possible à ceux du Repère Monde).

Usuellement, les marqueurs Aruco permettent un positionnement tridimensionnel (Figure 4.4 a)). Si l'on s'intéresse à un positionnement planaire en 2 dimensions pour localiser le robot, nous adopterons, pour nos marqueurs Aruco le référentiel *Marqueur*  $\mathcal{R}_{QR}$  proposé dans la Figure 4.4 b), qui omet l'axe vertical.

Pour les marqueurs des ensembles QRC1 ou QRC2, l'angle  $\theta_{QR}$  désignant la rotation du référentiel  $\mathcal{R}_{QR}$  par rapport à  $\mathcal{R}_M$  prend une des valeurs de l'ensemble  $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ , où les angles sont exprimés en radians. Les marqueurs de QRC3 ont tous la même orientation connue.

Ainsi donc, on peut exprimer la relation reliant le repère  $\mathcal{R}_{QR}$  à  $\mathcal{R}_M$  au travers de l'Équation 4.10 suivante :

$$\left[ O\mathcal{R}_{QR} |_{\mathcal{R}_M} \right] = \begin{bmatrix} R(\alpha) & T \\ 0^T & 1 \end{bmatrix} \cdot \left[ O\mathcal{R}_{QR} |_{\mathcal{R}_{QR}} \right] = MT_{\mathcal{R}_M |_{\mathcal{R}_{QR}}} \cdot \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (4.10)$$

$MT_{\mathcal{R}_M |_{\mathcal{R}_{QR}}}$  : Représente la matrice de transformation homogène de passage de  $\mathcal{R}_{QR}$  à  $\mathcal{R}_M$

$R(\alpha)$  : Représente la matrice de rotation de  $\mathcal{R}_{QR}$  par rapport à  $\mathcal{R}_M$  avec  $\alpha \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$

$T$  : Représente une translation qui exprime les coordonnées de  $OR_{QR}$  dans  $\mathcal{R}_M$



FIGURE 4.4 – Référentiels Aruco

En supposant que les murs soient parfaitement verticaux, que les couloirs 1 et 2 soient parfaitement perpendiculaires, nous adopterons dans la suite les incertitudes de positionnement absolu suivantes pour les marqueurs de l'environnement, dans notre carte. Les hypothèses que nous venons d'énoncer induisent qu'il n'existe pas d'erreur angulaire.

- Pavé centré incertitude marqueur selon l'axe x de  $\mathcal{R}_M$  :  $w_{QR |_{\mathcal{R}_M}}([x]) = 1$  cm
- Pavé centré incertitude marqueur selon l'axe y de  $\mathcal{R}_M$  :  $w_{QR |_{\mathcal{R}_M}}([y]) = 1$  cm
- Pavé incertitude de rotation du référentiel  $\mathcal{R}_{QR}$  par rapport à  $\mathcal{R}_M$  :  $w_{QR |_{\mathcal{R}_M}}([\theta_{QR}]) = 0$  rad

Incertitude 2 : Incertitudes de positionnement des marqueurs Aruco

Le cadre expérimental de notre étude étant précisé nous allons maintenant présenter comment nous allons prendre en compte les incertitudes impactant le calcul de la pose du robot.

## 4.4 La plateforme robotique : Vue matérielle

Dans cette partie nous allons détailler le fonctionnement et l'organisation tant matérielle que spatiale des différents éléments de la plateforme robotique. De plus, lorsqu'ils sont impliqués dans le calcul de la pose du robot, nous évaluerons les incertitudes maximales qu'ils peuvent garantir c'est à dire la dimension de la boîte au sein de laquelle se trouve les mesures que nous manipulons.

### 4.4.1 Organisation Matérielle

Nous disposons d'un robot Pioneer P3-Dx pour réaliser notre mission de Patrouille. C'est un robot, très robuste, développé historiquement par la société MobileRobots<sup>TM</sup> devenue maintenant Adept<sup>TM</sup>.

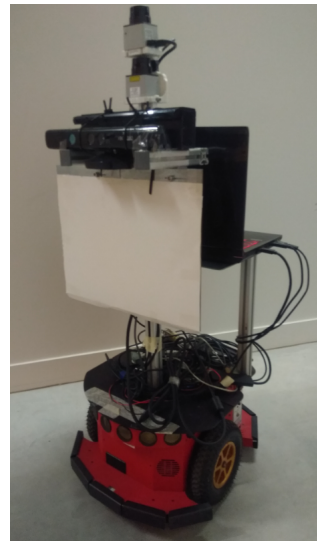
Il est composé d'une base différentielle mobile (Figure 4.5 a)) comportant 2 roues motrices, une roue folle, une ceinture de capteurs à ultrasons, et de capteurs de contact. Cette plateforme est donc sous-actionnée et non-holonome.

Comme le montre la Figure 4.5 b), nous avons adjoint à cette base une paire de capteurs laser montés tête-bêche au sommet d'un mat. Ce dernier supporte aussi, sur sa face frontale, une Kinect-1<sup>©</sup> qui sera simplement utilisée comme caméra sans faire appel à son capteur de profondeur pour la lecture des QR-Codes verticaux, et une Kinect-2<sup>©</sup>, qui sera de la même façon employée pour analyser les QR-Codes placés au plafond.

Enfin, à l'arrière de ce mat, on trouve une tablette supportant le PC-embarqué qui sera employé pour superviser la mission.



a) Base du Pioneer P3-DX



b) Base et éléments ajoutés

FIGURE 4.5 – Le robot patrouilleur

Le poids de la base du robot est de 9 kg pour une charge utile de 17 kg. Une large partie de cette masse provient des 3 batteries au plomb 12V dont est dotée la plateforme, ce qui lui confère une charge énergétique maximale théorique de 252 Wh. Avec l'ensemble des modifications apportées et le PC-embarqué le poids du robot atteint 25 kg. Nous nous trouvons donc pratiquement en limite de la charge utile acceptable. Sa vitesse maximale est de 1.2 m/s ce qui est trop important dans un environnement intérieur relativement confiné et dynamique. Nous l'avons donc limitée lors de la configuration logicielle du robot à 0.78 m/s.

Le PC-embarqué est relié d'une part à deux cartes comportant chacune 2 relais commandables qui lui permettent de contrôler l'alimentation des capteurs laser et Kinect depuis le PC-embarqué. D'autre part il est aussi relié au processeur de contrôle de bas-niveau du robot par le biais d'une liaison physique USB et d'un mécanisme d'échange de données client-serveur. Enfin il recueille aussi les informations renvoyées par un wattmètre embarqué permettant de suivre la consommation énergétique du robot.

On retrouve l'ensemble de ces éléments dans la représentation de l'organisation matérielle présentée ci-après en Figure 4.6.

On y trouve en blanc les dispositifs alimentés par la batterie du PC, en noir ceux reliés aux batteries robot, et enfin en gris ceux qui sont connectés aux deux sources d'énergie (USB et puissance). Enfin les pointillés identifient les composants qu'il est possible de déconnecter via les cartes relais.

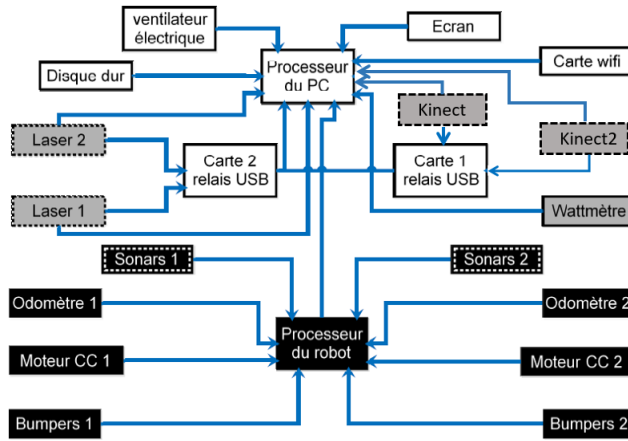


FIGURE 4.6 – Organisation matérielle du robot patrouilleur (d'après [112])

Après avoir présenté globalement l'organisation matérielle du robot patrouilleur, présentons maintenant son modèle cinématique.

#### 4.4.2 Modèle cinématique

Le modèle cinématique en posture est celui qui est couramment employé pour les robots unicycles de type Pioneer-P3Dx. Il va nous être utile pour pouvoir commander le robot et modéliser sa dérive odométrique. Il est construit sous l'hypothèse de non glissement et de non dérapage des roues.

On considère un robot évoluant dans un repère cartésien Monde  $\mathcal{R}_M$  et dont la position, à un instant  $t$ , est repérée par ses coordonnées cartésiennes  $(x,y)$ , et son orientation  $\theta$  définissant ainsi sa pose  $P_R(t)$ .

Classiquement on peut relier les vitesses angulaires et linéaires des robots des roues du robot, conformément aux notations de la Table 4.2 et de la Figure 4.7 par le système d'équations (4.11) :

$$\begin{cases} v_R = r \cdot \dot{\varphi}_R \\ v_L = r \cdot \dot{\varphi}_L \end{cases} \quad (4.11)$$

Symbole	Unité	Valeur Pioneer	Signification
$\dot{\varphi}_R, \dot{\varphi}_L$	rad/s	-	Vitesse angulaire roue droite (R) et gauche (G)
$v_R, v_L$	m/s	-	Vitesse linéaire roue droite (R) et gauche (G)
$(x,y)$	m	-	Position du robot dans $\mathcal{R}_M$
$\theta$	rad	-	Orientation du robot dans $\mathcal{R}_M$
$v$	m/s	-	Vitesse linéaire du robot
$\omega$	rad/s	-	Vitesse angulaire du robot
$r$	m	0.095	Rayon des roues
$b$	m	0.33	Distance entre-roues
$\rho$	m	-	Rayon de courbure

TABLE 4.2 – Paramètres du modèle cinématique d'un robot unicycle

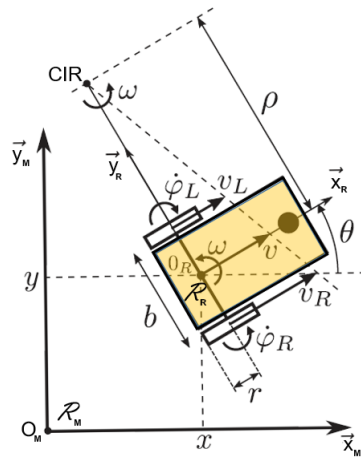


FIGURE 4.7 – Structure cinématique d'un robot unicycle (d'après [25])

En fonction des vitesses de rotation de chacune des roues, le robot tourne autour de son **Centre Instantané de Rotation** (CIR), passant par l'axe des 2 roues, selon un rayon de courbure  $\rho$ .

La vitesse linéaire et angulaire du robot sont alors reliées par la relation suivante (Equation 4.12) :

$$\begin{cases} v = \frac{v_R + v_L}{2} \\ \omega = \frac{v_R - v_L}{b} \end{cases} \quad (4.12)$$

Enfin on peut exprimer temporellement le lien entre la pose du robot, sa vitesse linéaire et angulaire, sa localisation courante et son orientation selon le modèle cinématique dit de posture [50]  $\dot{P}_R = (\dot{x}, \dot{y}, \dot{\theta})^T$ , donné par le système d'Équations 4.13.

$$\begin{cases} \dot{x} = v \cdot \cos \theta \\ \dot{y} = v \cdot \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (4.13)$$

Pour obtenir le modèle dynamique on peut faire appel classiquement [192] aux méthodes de Newton-Euler [235] et d'Euler-Lagrange [12]. Ce modèle bien que plus rigoureux reste complexe à prendre en compte car il nécessite l'identification de nombreux paramètres tels que les masses du robot et des roues, les moments d'inertie, ainsi la modélisation électrique des moteurs à courant continu, sans oublier celle du roulement avec glissement longitudinal et latéral et du contact roue-sol. Compte tenu de tous ces éléments nous nous limiterons donc au simple modèle cinématique du robot unicycle dans la suite de ce document.

Abordons maintenant l'organisation spatiale des différents composants du robot et son impact sur la précision de localisation qu'il sera possible d'atteindre.

### 4.4.3 Organisation spatiale

L'organisation spatiale des capteurs (Figure 4.8) que nous avons ajoutés va avoir une légère influence sur la précision du calcul de la pose du robot. En effet puisque la base du robot est de fabrication industrielle nous pouvons admettre que les sonars qui s'y trouvent nativement sont affectés d'erreurs de positionnement négligeables.

En revanche il n'en est absolument pas de même pour tous les éléments que nous avons ajoutés à cette base, que ce soit le mât, les capteurs laser montés tête-bêche ou les Kinects.



Ces matrices de transformation supposent que :

- Les mats supports ne sont affectés d’aucune erreur statique de rotation.
- Les 2 lasers positionnés tête-bêche l’un au dessus de l’autre sont exactement superposés verticalement selon l’axe z (non représenté).
- Les mesures de distance du Laser 2 obtenues dans le repère  $\mathcal{R}_{Las2}$ , dirigé vers l’arrière du robot sont préalablement exprimées dans le repère du Laser 1  $\mathcal{R}_{Las1}$ , dirigé vers l’avant dont le référentiel est orienté de la même façon que le Référentiel Robot  $\mathcal{R}_R$ , à l’aide d’une matrice de rotation d’angle égal à  $\Pi$ . Donc dans la suite le référentiel Laser  $\mathcal{R}_{Las}$  correspondra au référentiel du Laser 1  $\mathcal{R}_{Las1}$ .

La Table intervalliste d’incertitudes 3 suivante précise les incertitudes que nous avons estimées sur les éléments ajoutés.

Axe	$\mathcal{R}_C$	$\mathcal{R}_{Ref}$	Type	$\bar{X}$ (m)	$\underline{X}$ (m)	[w] (m)
X	$\mathcal{R}_{Las}$	$\mathcal{R}_R$	Centrée	0.025	0.015	0.01
Y	$\mathcal{R}_{Las}$	$\mathcal{R}_R$	Centrée	0.0035	-0.0035	0.007
X	$\mathcal{R}_{Kin}$	$\mathcal{R}_R$	Centrée	0.19	0.18	0.01
Y	$\mathcal{R}_{Kin}$	$\mathcal{R}_R$	Centrée	0.0035	-0.0035	0.007
X	$\mathcal{R}_{Kin2}$	$\mathcal{R}_R$	Centrée	0.08	0.09	0.01
Y	$\mathcal{R}_{Kin2}$	$\mathcal{R}_R$	Centrée	-0.0725	-0.0785	0.007

Incertitude 3 : Table intervalliste d’incertitude de positionnement spatial

Cette table a été établie en supposant que :

- La précision de montage des capteurs laser et Kinect n’induit pas d’incertitude en rotation
- L’erreur de positionnement du centre de rotation est de 0. mm selon les axes X et Y de  $\mathcal{R}_R$ .
- L’erreur de positionnement du mât est de 2 mm et 0.5 mm selon les axes X et et Y de  $\mathcal{R}_R$  respectivement.
- L’erreur de positionnement des capteurs est de 2.5 mm selon les axes X et et Y de  $\mathcal{R}_R$  respectivement.

Après avoir précisé l’organisation spatiale de notre système robotisé nous allons maintenant balayer l’ensemble de ces capteurs et préciser pour chacun d’eux, lorsque cela est nécessaire, les incertitudes de mesure associées.

#### 4.4.4 Capteurs

Pour présenter les capteurs de notre robot nous commencerons par les capteurs interoceptifs, puis passerons aux proprioceptifs et enfin nous terminerons par les extéroceptifs qui, pour la plupart, sont centraux dans le calcul de la pose du robot.

##### 4.4.4.1 Wattmètre embarqué

Nous disposons sur notre robot d’un seul capteur intéroceptif, un wattmètre embarqué de type Yocto-Watt (Figure 4.10). Il a été ajouté en série avec les batteries pour pouvoir suivre en permanence, tout au long d’une mission la consommation énergétique du robot. Il permet de mesurer, via une liaison USB avec le PC embarqué, le courant, la tension, la puissance, la durée de mesure et l’énergie consommée (avec une précision de 1 mWh).



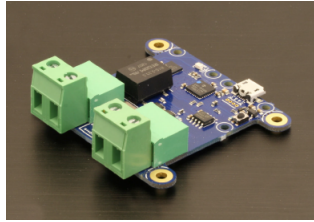


FIGURE 4.10 – Le wattmètre embarqué Yocto-Watt

Il n'est bien évidemment d'aucun intérêt pour localiser le robot. En revanche il joue un rôle très important pour la gestion de la performance énergétique en permettant de comparer celle prédite par la méthodologie PANORAMA et celle effectivement réalisée. Il concourt donc au calcul et à la mise à jour de la Marge énergétique réelle.

#### 4.4.4.2 Capteurs de choc

Pour des raisons de sécurité en cas de choc inopiné le robot dispose d'une ceinture de capteurs de choc (*bumper*). Elle est constituée de deux banques de 6 bumpers situées à l'avant et à l'arrière du robot (Figure 4.11). Ils sont utilisés en cas de détection de choc pour stopper immédiatement le robot. Ces capteurs ne sont évidemment pas employés pour localiser le robot.

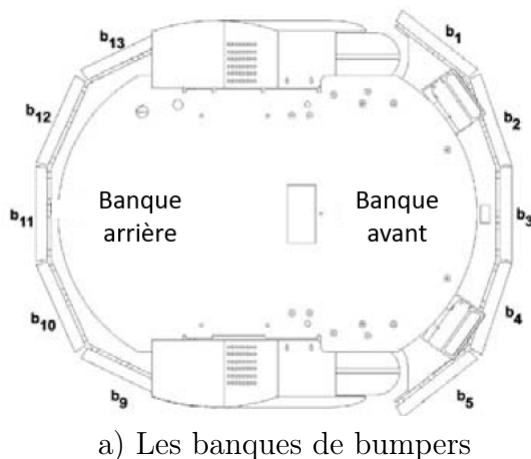


FIGURE 4.11 – Les détecteurs de choc

Passons maintenant à l'odomètre, le seul capteur proprioceptif que nous ayons à notre disposition. Il joue un rôle central dans la localisation du robot et la réalisation de la mission.

#### 4.4.4.3 Odomètre

Malgré sa dérive, l'odométrie joue très souvent un rôle central pour la localisation des robots en environnement intérieur où la navigation à l'estime est souvent employée. Sur la plateforme que nous utilisons, les seuls capteurs proprioceptifs dont nous disposons sont des encodeurs de type E203-500 placés sur l'axe de rotation des moteurs entraînant les roues du robot. Ce sont des encodeurs optiques à deux canaux en quadrature. Le décalage de  $90^\circ$  des 2 pistes l'une par rapport à l'autre permet de détecter le sens de rotation des roues. Ces encodeurs génèrent 500 fronts par tour d'axe du moteur ( $Nb_{front/tr}$ ). Ils mettent en œuvre un encodage dit "x4" permettant de détecter les fronts montants et descendants engendrés par la rotation. Enfin un réducteur de rapport de réduction  $R = 38.3 : 1$  est utilisé de façon à adapter la vitesse de rotation des moteurs à celles des roues et à augmenter le couple.

La documentation technique du robot [166] mentionne la définition d'un paramètre  $Ticksmm$  (Equation 4.17) représentant le nombre de fronts détectés par millimètre parcouru par une roue de diamètre  $\varnothing_{roue} = 190 \text{ mm}$ . Ce paramètre de calibration peut être modifié de façon à ce que la distance réelle parcourue corresponde effectivement à celle mesurée par le robot. On peut constater facilement au vu de la valeur numérique de ce paramètre que la discrétisation du comptage du nombre de fronts n'engendre qu'une erreur théorique totalement négligeable (Incertitude 4) sur la mesure de la distance parcourue par une roue. C'est à la fois la force de l'odométrie puisqu'elle permet de détecter des déplacements infinitésimaux du robot, mais aussi son talon d'Achille, puisque les différences les plus infimes (diamètre roues, micro-glissement, etc...) détérioreront la qualité des mesures proprioceptives de manière cumulative.

$$Ticksmm = \frac{4.Nb_{front/tr}.R}{\Pi.\varnothing_{roue}} = 128 \text{ fronts/mm} \quad (4.17)$$

- Incertitude encodeur :  $\delta_{Encodeur} = 7.8 \cdot 10^{-6}m$

Incertitude 4 : Incertitude liée à l'encodeur

Ainsi donc, en décomptant dans le temps, à partir d'une position connue de référence, la distance parcourue par chacune des roues et en prenant en compte leur sens de rotation il est alors possible de connaître la vitesse de rotation de chacune d'elles. En utilisant le modèle cinématique du robot on peut alors estimer la pose courante du robot par rapport à celle de référence.

Ce calcul est automatiquement réalisé par le contrôleur de bas niveau du robot, qui, à partir des informations issues des encodeurs, renvoie la pose estimée du robot  $\left( PO_R(t) \right)^T = \left( x_R(t), y_R(t), \theta_R(t) \right)^T$  par rapport à une situation de référence initiale choisie  $\left( PO_R(0) \right)^T = \left( 0, 0, 0 \right)^T$ , l'axe x étant pointé devant lui (Figure 4.12). L'orientation de ce référentiel correspond à celui choisi pour le référentiel Robot  $\mathcal{R}_R$ .

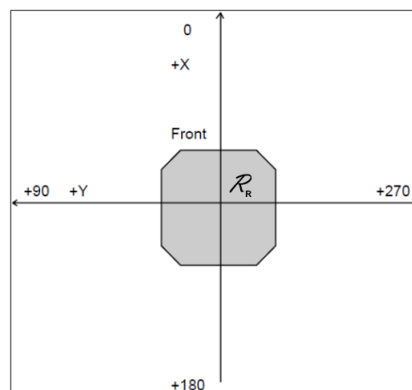


FIGURE 4.12 – Pioneer P3-Dx : Système de coordonnées internes - Référentiel Robot  $\mathcal{R}_R$

Il n'en demeure pas moins qu'en raison de nombreux phénomènes (diamètre différent des roues, glissements, etc.) que nous détaillerons dans la suite, la précision de cette estimation va rapidement se détériorer. Deux paramètres de calibration sont mis à la disposition de l'utilisateur pour limiter ce phénomène. Le *Driftactor* permet, si nécessaire, de prendre en compte

la différence de diamètre des 2 roues. Il est utilisé pour compenser la dérive du robot lorsqu'il doit se déplacer en ligne droite. Le paramètre *Revcount* est utilisé pour limiter l'erreur de cap, il décompte le nombre de fronts enregistrés lors d'un demi-tour du robot. Il dépend aussi d'un grand nombre de facteurs, l'empattement des roues, leur usure, la charge transportée, la surface d'évolution, etc...

Les paramètres de calibration constructeur sont présentés dans la Table 4.3.

Ticksmm = 128 (front/mm)	Driftfactor = 0	Revcount = 16570 (front)
--------------------------	-----------------	--------------------------

TABLE 4.3 – Paramètres de calibration constructeur du Pioneer P3-Dx

Nous avons choisi de ne pas modifier la valeur de ces paramètres directement au niveau du contrôleur bas-niveau du robot. Comme nous le verrons dans la suite nous le ferons logiquement au niveau de l'architecture de contrôle.

Compte tenu de ses caractéristiques, l'odomètre peut bien évidemment être mis en œuvre sur l'ensemble de la carte, puisqu'il ne fait que compter des tours de roues. Cependant, bien évidemment, en raison des inévitables glissements pouvant affecter la rotation des roues il faudra veiller à caractériser la dérive de localisation engendrée. Nous y reviendrons plus longuement dans la suite du document.

Après avoir présenté le seul capteur proprioceptif présent sur la plateforme nous allons maintenant détailler l'ensemble des capteurs extéroceptifs présents.

#### 4.4.4.4 Ultrasons

Les ultrasons dont dispose la plateforme robotique ne sont absolument pas utilisés pour contribuer à la localisation du robot mais uniquement à des fins sécuritaires. Ils sont répartis en 2 banques (frontale et arrière voir Figure 4.13 a)) composées chacune de 8 capteurs à ultrasons (Figure 4.13 b)) de la série 6500 de Polaroid initialement développés pour la mise au point de caméras, et mettant en œuvre des transducteurs électrostatiques. La mesure de distance fait appel au calcul du temps de vol (*Time Of Flight*) séparant l'émission d'une onde ultra-sonore de sa réception après réflexion sur un obstacle. Ils permettent de détecter les surfaces translucides telles que les vitres mais sont d'une portée relativement limitée (de 0.1 m à 4 m).

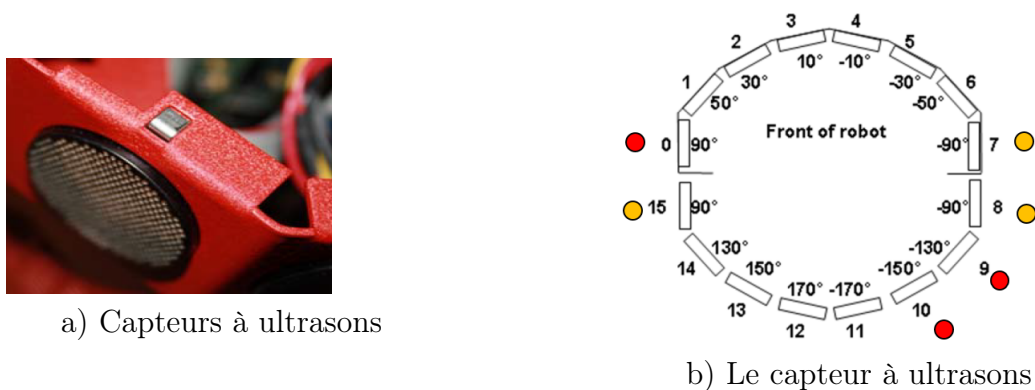


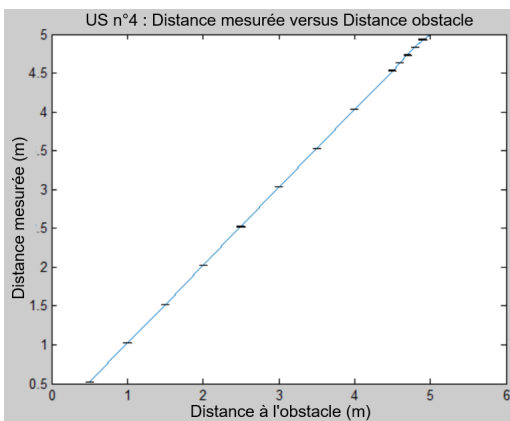
FIGURE 4.13 – Les banques de capteurs à ultrasons du Pioneer P3-DX

Ils sont très sensibles aux échos secondaires venant perturber une mesure. C'est pourquoi sur le Pioneer P3-DX l'insonification est multiplexée et se fait capteur par capteur, banque après banque, de la gauche vers la droite, à une fréquence réglable fixée par défaut à 25 Hz. Cela veut

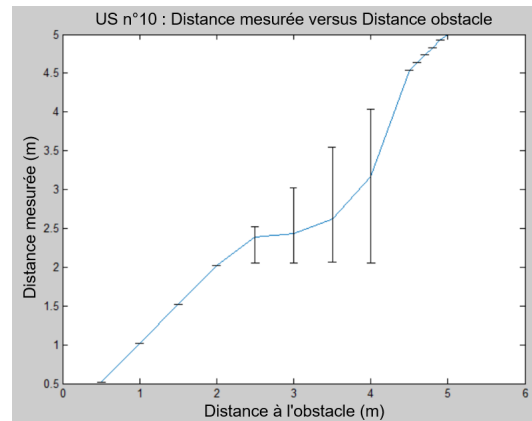
donc dire que toutes les 40 ms un sonar effectue une mesure dans une direction donnée, et que ce même capteur est à nouveau sollicité 320 ms plus tard. Grâce aux 2 banques disponibles on obtient donc des informations planaires sur la localisation d'obstacles sur  $360^\circ$ .

Même si les sonars ne sont sollicités que pour des fins sécuritaires, nous avons tout de même caractérisé la précision de chacun d'entre eux en statique c'est à dire sans que le robot ne se déplace. Pour ce faire nous avons positionné face à chacun des ultrasons un obstacle que nous avons déplacé tous les 50 cm entre 0.5 m et 4.5 m, puis tous les 10 cm entre 4.5 m et 5 m. Nous avons effectué 50 relevés de mesure pour chaque position et notre réalité terrain était connue à 0.5 cm près. Le robot nous renvoie la distance entre l'obstacle le plus proche et le capteur testé.

Cette expérimentation nous a permis d'obtenir 2 types de relevés. D'une part, comme le montre la Figure 4.14 a), pour 11 sonars sur 16 une bonne corrélation entre la distance mesurée et la réalité terrain. D'autre part, pour 5 d'entre eux, comme le montre la Figure 4.14 b), un comportement localement (ronds oranges 4.13 b)) fortement non linéaire (ronds rouges 4.13 b)), traduisant un dysfonctionnement du dispositif pour certaines plages de mesure.



a) Réponse typique d'un capteur à ultrasons



b) Capteur à ultrasons défectueux

FIGURE 4.14 – Caractérisation des US : Exemples de relevés expérimentaux

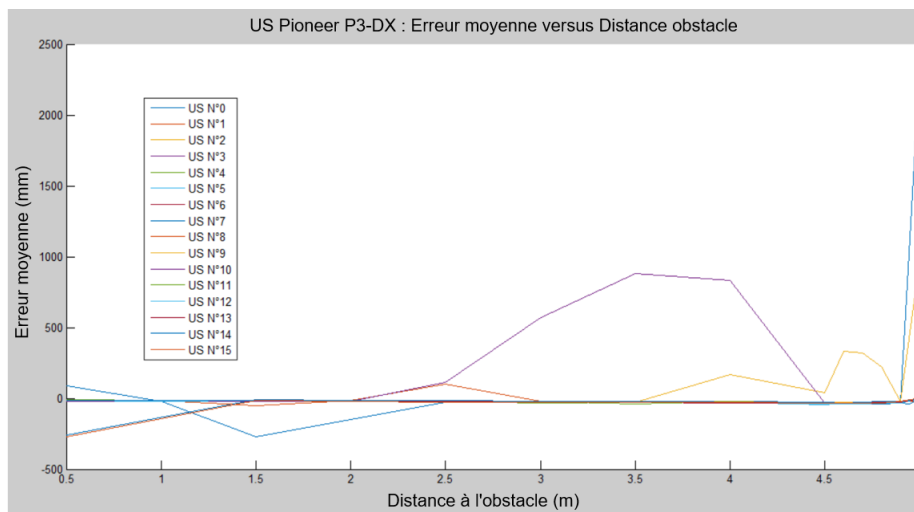


FIGURE 4.15 – Pioneer P3-Dx : Erreur de mesure de distance des capteurs à ultrasons

Enfin la Figure 4.15 présente pour l'ensemble des ultrasons l'erreur moyenne relevée pour chacune des distances évaluées. On constate bien que 5 courbes d'erreur se démarquent très

nettement des autres (US n° 0, 7, 8, 9, 10, 15) avec des erreurs de mesures plus ou moins localisées pouvant atteindre près de 2.5 m ! Si l'on analyse leur localisation on constate que nous sommes myope des 2 côtés dans la zone médiane du robot et sur son côté arrière droit. Fort heureusement ce dysfonctionnement n'affecte pas directement la face avant du robot, critique pour la mise en œuvre de l'évitement d'obstacle lors du déplacement normal de ce dernier. Pour les sonars fonctionnant correctement on mesure une erreur moyenne de -0.024 m, et l'erreur maximale observée pour une mesure de distance est de -0.042 m.

Conformément à la documentation Pioneer nos relevés expérimentaux font apparaître que les capteurs à ultrasons sont capables de détecter des obstacles au delà de 4 m avec une précision de quelques millimètres. Ces résultats sont corroborés par les travaux présentés dans [48] qui ont par ailleurs déterminé, aussi en statique, que la sensibilité de ce capteur était de 1 mm et que la distance minimale de détection observée était légèrement différente de celle annoncée.

Compte tenu des caractéristiques qui leur permettent de détecter tous types de matériaux y compris les vitres, les capteurs à ultrasons sont donc utilisables sur l'ensemble de la carte de notre environnement. Qui plus est, ils sont mêmes indispensables dans les zones où les autres capteurs ne peuvent renvoyer des informations pertinentes (Figure 4.16 ). La zone 1 de la halle est fortement non structurée et dynamique, dans la zone 2 du couloir 1 qui simule un environnement vitré, dans la zone 3 où les lasers renvoient des informations erronées en raison de la couleur et texture de l'ascenseur, dans la zone 4, se trouvant à proximité de la vanne 2 qui comporte des obstacles trop bas (canapé - fauteuils) pour être détectés par nos lasers.

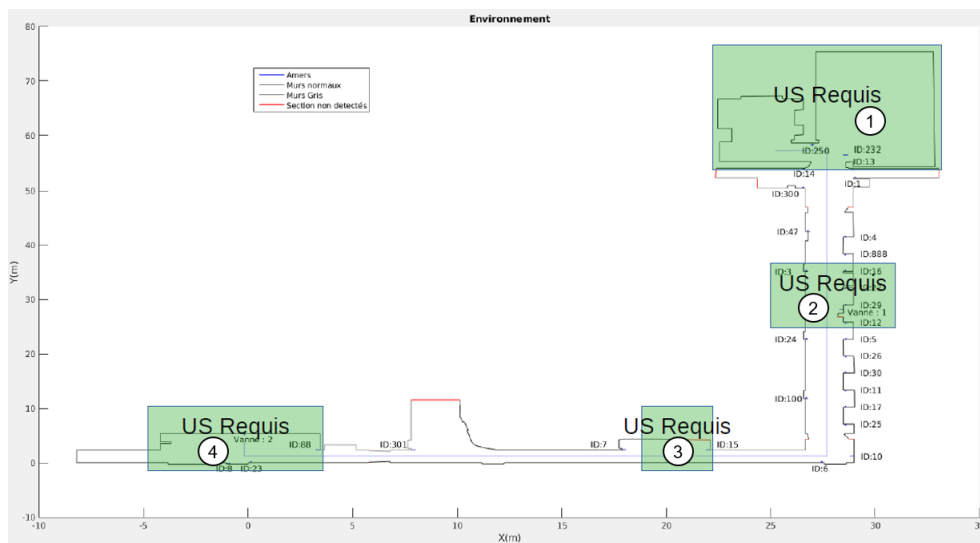


FIGURE 4.16 – Carte de l'environnement : Zones où les capteurs à ultrasons sont indispensables

Après avoir évalué la précision de mesure des capteurs ultrasons de notre robot, nous allons maintenant nous intéresser aux capteurs impliqués dans sa localisation extéroceptive, en commençant par les lasers.

#### 4.4.4.5 Laser

Sur notre plateforme nous utilisons donc 2 capteurs laser Hokuyo URG-04LX 2D (Figure 4.17 a)) montés tête-bêche au sommet d'un mat. Nous allons commencer par présenter leur principe de fonctionnement ainsi que la signature 360° qu'ils génèrent avant de nous attarder sur la précision de leurs mesures et sur les facteurs qui peuvent impacter leur utilisation.

Contrairement aux capteurs à ultrasons, la mesure de distance ne s'appuie pas sur le calcul d'un temps de vol, mais sur celui de la modulation d'amplitude (*Amplitude Modulated Continuous Wave*). Dans ce cas une lumière laser modulée en amplitude est émise et on mesure sa différence de phase avec la lumière réfléchiée pour déduire la distance à un obstacle. Au sein du dispositif le rayon laser est tout d'abord réfléchi sur un miroir rotatif de façon à obtenir une représentation planaire de l'environnement sur  $240^\circ$  (Figure 4.17 b)) avec un pas angulaire d'environ  $0.36^\circ$  en 100 ms, une résolution de 1 mm, et une distance maximale de mesure de 4 m. Cela correspond à un relevé planaire de 683 points. Le constructeur annonce une précision de  $\pm 1$  cm entre 2 cm et 1 m, puis de  $\pm 1\%$  de la distance mesurée au delà, soit  $\pm 4$  cm à 4 m.

Pour les 2 capteurs laser qui ont été placés tête-bêche on obtient donc une couverture que nous supposons planaire, alors qu'ils sont bien évidemment l'un au dessus de l'autre, sur  $360^\circ$  comportant 1022 points, 511 pour la capteur dirigé vers l'arrière, et autant pour celui dirigé vers l'avant. Une rotation de  $0.7^\circ$  observée du laser avant a été corrigée par logiciel.

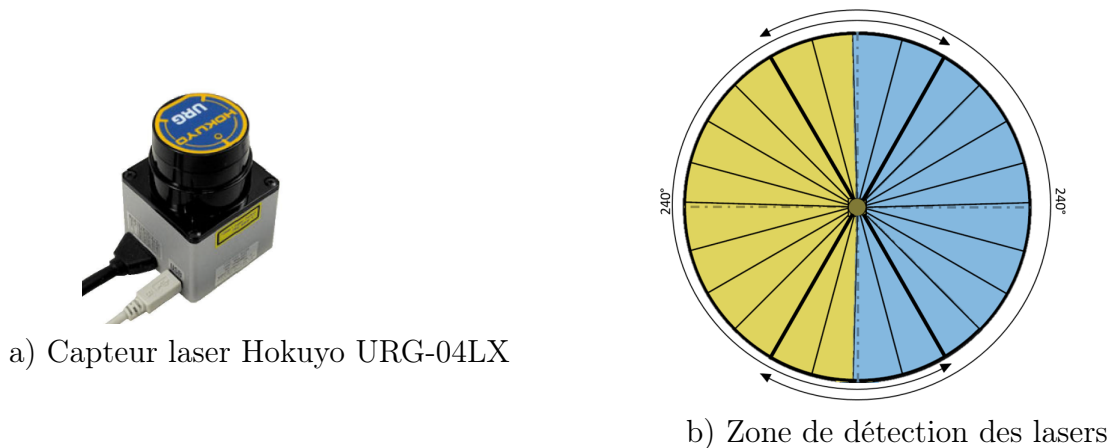


FIGURE 4.17 – Les capteurs laser du Pioneer P3-DX

Nous avons caractérisé la précision de nos capteurs laser. Pour ce faire nous avons relevé la valeur de la mesure de distance obtenue face à un mur de béton blanc. Nous avons déplacé notre capteur tous les 10 cm de 0.5 m à 1 m, puis tous les 50 cm jusqu'à une distance de 3.5 m, puis à nouveau tous les 10 cm jusqu'à 4 m. Pour chaque distance nous avons effectué environ 10000 mesures pour une réalité terrain connue à 0.5 cm près. Ces relevés, effectués statiquement sans que le robot ne bouge, ont permis d'obtenir la courbe de réponse de la figure 4.18 pour le laser avant. Une courbe présentant sensiblement les mêmes caractéristiques a été obtenue pour le laser arrière.

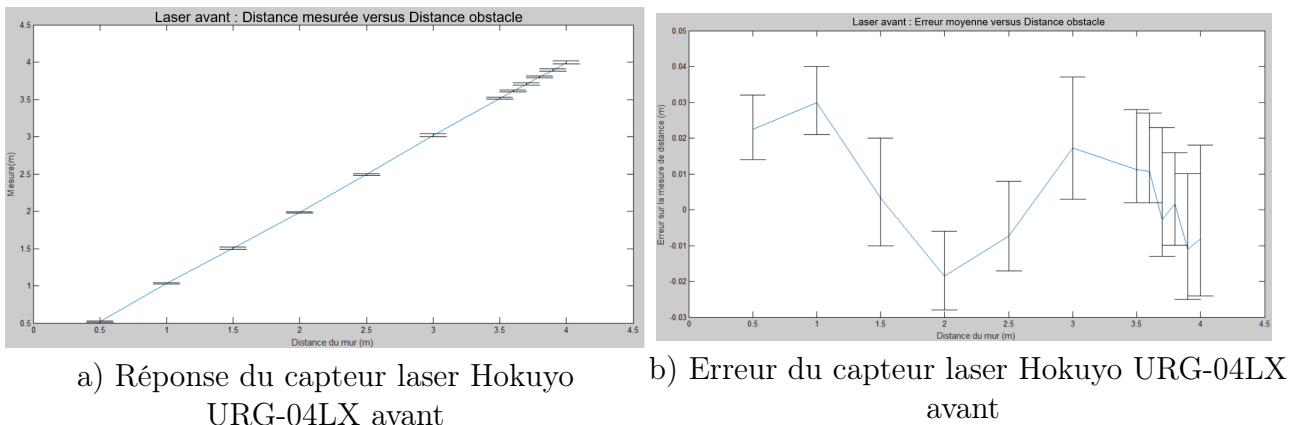


FIGURE 4.18 – Courbes de réponse type des capteurs lasers du Pioneer P3-DX

On constate sur la Figure 4.18 a) que la courbe de mesure obtenue est pratiquement linéaire. L'analyse de l'erreur moyenne présentée en 4.18 b) montre une erreur minimale de -0.028 m, maximale de 0.04 m et moyenne de 0.37 cm. On constate donc que l'erreur observée se trouve bien inférieure ou égale à l'incertitude maximale constructeur, même si cette valeur n'est pas obtenue pour la distance de 4 m. Nous retiendrons donc pour la suite une erreur de mesure de distance égale à 4 cm.

Mais les capteurs lasers présentent d'autres variabilités qui ont été analysées pour l'Hokuyo URG-04LX dans [132] et [173]. L'une d'elles concerne son temps de chauffe. En effet, l'étude présentée dans [173] montre que la stabilité de la mesure n'est obtenue qu'au bout de 90 minutes, avec un intervalle de variation de 0.3% de la distance mesurée à 2 m. Cette courbe de réponse décroît avec le temps avant de se stabiliser. Donc si la mission est courte, et si le laser n'a pas atteint son régime permanent, les mesures de distance peuvent être surévaluées d'environ 1 cm [132].

Les mesures de ce capteur sont aussi influencées par l'atmosphère lumineuse ambiante. Dans le noir les mesures sont minorées d'environ 5 mm [132], il faut donc que le robot évolue avec un éclairage ambiant de bonne qualité. Les expérimentations présentées dans [132] [173] montrent l'influence de la texture (aluminium, acier, bois, etc..) de l'environnement, de sa couleur, et de sa brillance sur les mesures effectuées. Si l'on couple ces facteurs avec l'impact de l'angle d'incidence on peut atteindre jusqu'à 100% de mesures erronées pour l'aluminium. Sans compter qu'un relevé peut comporter des points virtuels (*mixed-pixel*), ne correspondant à aucun objet réel de l'environnement si ce dernier présente une alternance de zones proches et éloignées [173].

A ces sources de variabilité des mesures il faut aussi ajouter d'autres facteurs liés au robot lui-même. D'une part les lasers sont placés sur un socle au sommet d'un mat. Si comme nous l'avons déjà dit une rotation autour de l'axe z peut être facilement compensée, il peut aussi y avoir une légère erreur d'horizontalité entraînant une rotation d'angle  $\alpha$  autour de l'axe y. Dès lors le rayon laser va potentiellement parcourir un trajet  $\frac{D}{\cos \alpha}$  supérieur à la distance de référence D. Cependant pour un angle  $\alpha$  faible l'impact reste limité.

D'autre part, puisque le relevé planaire s'effectue en 100 ms avec un robot en mouvement, chaque mesure devrait prendre en compte la trajectoire du robot et ses variations de position et de cap, pour repositionner correctement tous les points mesurés pendant la période de scrutation. Pour des vitesses faibles, ce qui est encore le cas dans notre application, on peut considérer que toutes les mesures ont été obtenues en fin de balayage. Pour des véhicules se déplaçant à des vitesses supérieures, comme les applications impliquant des voitures par exemple, il devient alors indispensable de prendre en compte le déplacement du véhicule [1], même si tous les travaux relevant de ce contexte applicatif ne le font pas systématiquement [24].

On peut donc conclure que le capteur laser présente l'avantage de fournir un nombre important d'informations sur l'environnement du robot sur une large zone planaire. Cependant les mesures effectuées sont, somme toute, soumises à l'influence de nombreux facteurs qui peuvent induire des erreurs de mesures non négligeables. Compte tenu de ce constat, pour une mesure de distance effectuée par un laser dans un environnement favorable (texture, couleur, luminosité), nous retiendrons le pavé d'incertitudes suivant (Incertitude 5) exprimé dans le référentiel capteur en supposant que l'erreur maximale de mesure d'une distance est de  $\Delta D = 4 \text{ cm}$ , pour un rayon laser présentant un angle  $\theta_{Las}$  avec l'axe X du repère capteur.

$$\begin{aligned}\Delta X &= \Delta D. \cos \theta_{Las} \\ \Delta Y &= \Delta D. \sin \theta_{Las} \\ \Delta \theta &= 0.36^\circ\end{aligned}$$

#### Incertitude 5 : Incertitudes de Mesure des capteurs laser

Par ailleurs, même si les lasers peuvent être utilisés dans l'absolu sur tout notre environnement, la qualité des informations renvoyées peut être entachée d'erreurs. La carte d'utilisation suivante (Figure 4.19) montre qu'ils peuvent être perturbés par la couleur des murs et qu'ils ne détectent pas des obstacles trop bas ainsi que les vitres.

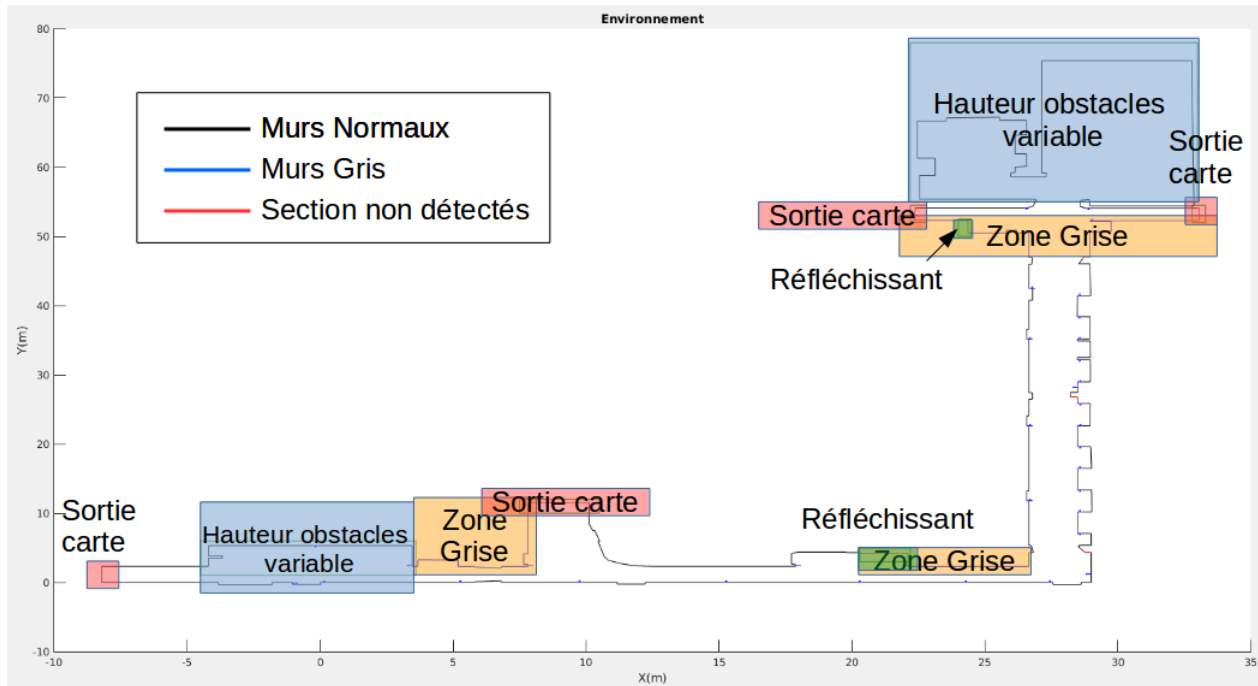


FIGURE 4.19 – Carte de l'environnement : Zones impactant les mesures laser

Après avoir évalué la précision de mesure des capteurs laser de notre robot, nous allons maintenant nous intéresser aux autres capteurs extéroceptifs dont nous disposons, des caméras Kinect.

#### 4.4.4.6 Kinect 1 et marqueurs Aruco

Nous disposons sur notre plateforme robotique d'un capteur Xbox Kinect v 1.0 [202] pour analyser les marqueurs Aruco verticaux. C'est un capteur très largement utilisé qui, pour un coût réduit, permet d'adresser un large panel d'applications telles que l'acquisition et la modélisation de scènes 3-D, le suivi de personnes, la reconnaissance de gestes, mais aussi de nombreuses applications robotiques comme celles liées à la navigation. La première génération que nous employons ici et qui date des années 2010 est un dispositif dit à lumière structurée, composé de 3 principaux éléments (Figure 4.20). D'une part une caméra RGB de 640 x 480 pixels fonctionnant à la fréquence de 30 Hz. D'autre part un émetteur infra-rouge projetant un motif sur la scène observée. Enfin un capteur infra-rouge qui permet par triangulation de déterminer, pour chaque pixel, la distance à laquelle il se trouve de l'objectif. Enfin un moteur permet de contrôler le tilt du dispositif.

Dans notre application nous n'utilisons pas l'image de profondeur fournie par la Kinect. Elle ne constitue donc pour nous qu'une simple caméra. Nous l'employons pour détecter les



marqueurs Aruco disséminés dans l'environnement. C'est donc la performance de Localisation Aruco que nous avons donc cherché à caractériser plutôt que celle de la Kinect proprement dite. Nous avons effectué des mesures avec le robot à l'arrêt, donc statiquement, et avec le robot en mouvement, donc dynamiquement bien évidemment après avoir calibré au préalable notre caméra.

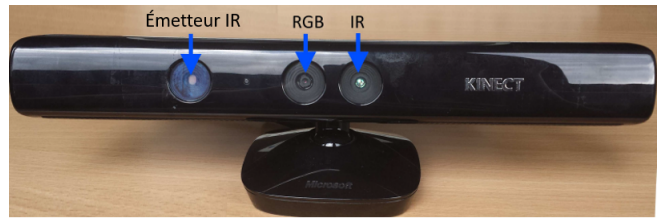
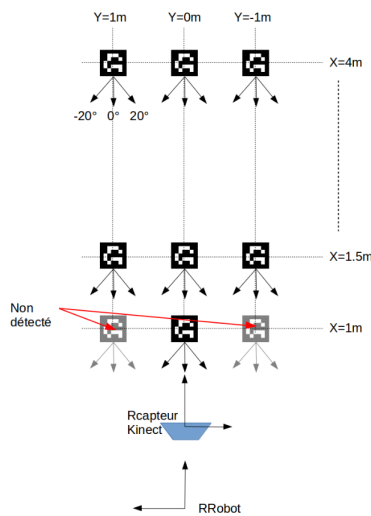
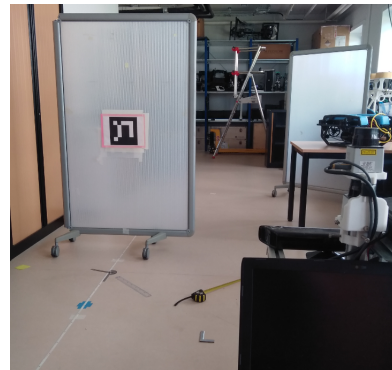


FIGURE 4.20 – Le capteur Kinect v 1.0 à lumière structurée

**A Caractérisation statique** Statiquement, le robot étant fixe nous avons déplacé un marqueur Aruco (Figure 4.21 b)) entre 1 m et 4 m tous les 50 cm selon 3 lignes droites, l'une faisant directement face à la Kinect et les 2 autres décalées de 1 m vers la gauche et la droite de la 1ère (Figure 4.21 a)). Pour chacune de ces positions nous avons effectué des mesures pour 3 orientations du marqueur, l'une où le marqueur est parallèle à la Kinect, et 2 autres où une rotation de  $\pm 20^\circ$  du marqueur a été effectuée. La réalité terrain a été obtenue à l'aide d'un décimètre à ruban et d'un rapporteur.



a) Schéma de principe de la caractérisation statique de Aruco



b) Caractérisation statique Aruco : Un exemple de mesure

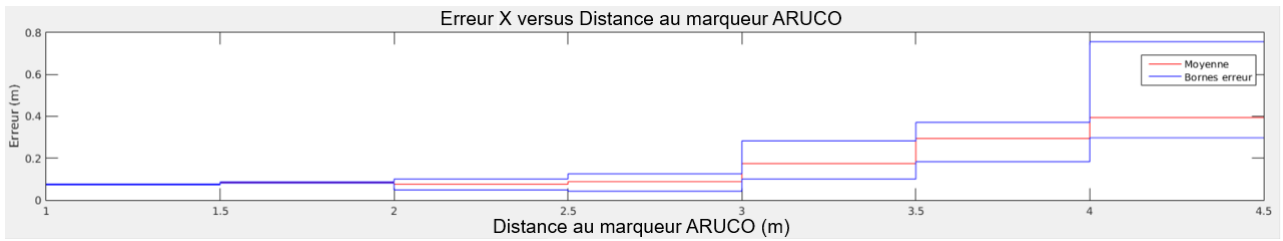
FIGURE 4.21 – Caractérisation statique des marqueurs Aruco sur la Kinect v.1

Les courbes de la Figure 4.22 représentent les erreurs de détection relevées en X, Y et  $\theta$  en fonction de la position du marqueur. Chacune d'elles se réfère en abscisse à la distance au capteur pour  $Y=0$  mais intègre les relevés d'erreurs pour toutes les positions et orientations envisageables. On constate que ces erreurs ne sont pas nécessairement centrées. Selon X l'erreur reste faible jusqu'à 2.5 m  $[-0.17, +0.12]$  mais qu'elle diverge ensuite fortement pour atteindre près de 0.75 m à 4 m. Selon Y l'erreur commence à diverger au delà de 1.5 m mais limitée au maximum à 0.15 m quelle que soit la distance du marqueur. Enfin l'erreur d'orientation reste limitée  $[-0.08, +0.18]$  jusqu'à 2.5 m mais peut diverger ensuite fortement positivement pour atteindre jusqu'à 0.7 rad. En conclusion globalement on peut énoncer qu'en statique les

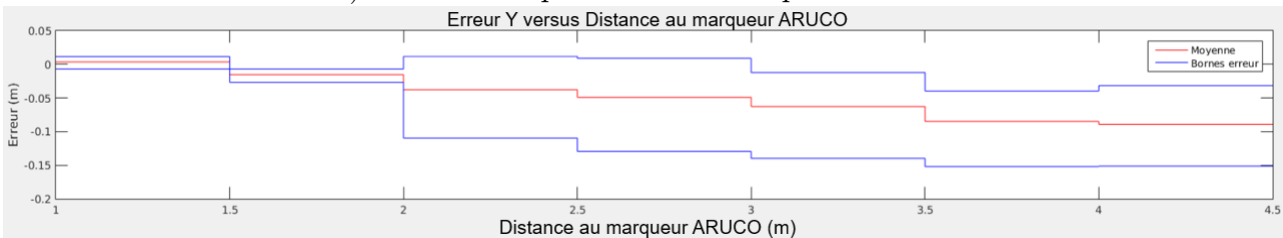
mesures effectuées avec nos marqueurs Aruco utilisés avec une Kinect V.1 peuvent fortement diverger au delà de 2.5 m, alors que l'erreur en Y reste limitée selon Y. Ces mesures seront traduites par 7 pavés d'incertitudes pour des distances allant de 1 m à 4 m par pas de 0.5 m. A titre d'exemple on trouve dans la table d'incertitudes suivante (6) 2 exemples de pavé, l'un à la distance d'un mètre, et l'autre à la distance de 2.5 m.

Distances au QR-Codes (m)	Incertitude X (m)	Incertitude Y (m)	Incertitude d'orientation (rad)
[1, 1.5]	[0.073, 0.076]	[-0.007, 0.012]	[-0.04, 0.07]
[1.5, 2]	[0.081, 0.087]	[-0.027, -0.007]	[-0.03, 0.05]
[2, 2.5]	[0.049, 0.101]	[-0.110, 0.012]	[-0.10, 0.15]
[2.5, 3]	[0.043, 0.126]	[-0.129, 0.009]	[-0.08, 0.19]
[3, 3.5]	[0.100, 0.283]	[-0.140, -0.012]	[-0.07, 0.68]
[3.5, 4]	[0.184, 0.371]	[-0.152, -0.040]	[-0.08, 0.69]
[4, 4.5]	[0.297, 0.756]	[-0.151, -0.032]	[-0.12, 0.11]

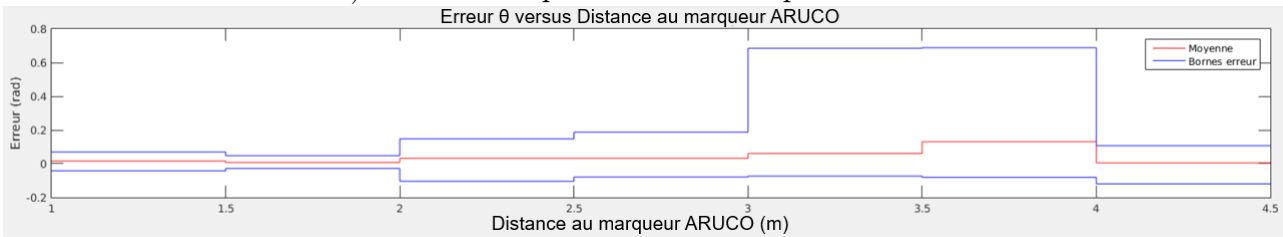
Incertitude 6 : Exemples de pavé d'incertitude Kinect/Aruco en statique



a) Erreur statique en X des marqueurs Aruco



b) Erreur statique en Y des marqueurs Aruco



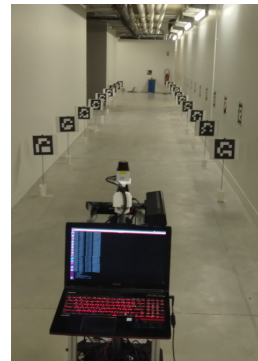
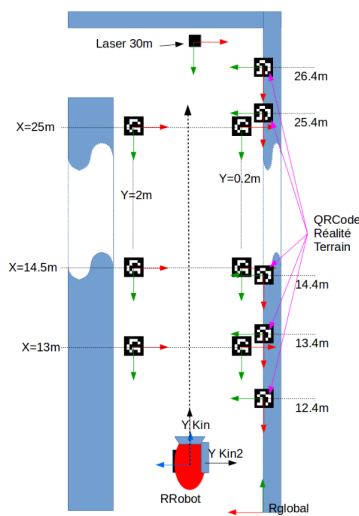
c) Erreur statique en  $\theta$  des marqueurs Aruco

FIGURE 4.22 – Caractérisation statique des marqueurs Aruco sur la Kinect v.1

**B Caractérisation dynamique** Les valeurs d'erreur que nous venons d'obtenir statiquement peuvent être impactées par le déplacement du robot qui peut induire des vibrations et un floutage des images Kinect traitées par Aruco. Le modèle dynamique d'erreur a été beaucoup plus compliqué à obtenir. Pour y parvenir nous avons tout d'abord choisi de faire déplacer notre robot en ligne droite, en boucle ouverte, à différentes vitesses (0.33, 0.4 et 0.56 m/s) dans une partie du couloir 2 de notre environnement. Les marqueurs que le robot devra détecter

avec la Kinect v.1, seront placés tous les 1.5 m de part et d'autre du chemin que le robot doit suivre. Lors des relevés nous analyserons les données issues de tous les marqueurs détectés par la Kinect et pouvant être traités par Aruco.

Pour pouvoir obtenir une réalité terrain nous avons apporté plusieurs aménagements à notre environnement et effectué plusieurs modifications sur notre plateforme. En premier lieu nous avons équipé un mur d'une partie du couloir 2 de notre environnement avec un ensemble de marqueurs Aruco placés tous les mètres. Ne pouvant évaluer la précision de la Kinect v.1 avec celle-ci, nous avons complété notre plateforme avec une Kinect v.2 qui fera face aux marqueurs du mur et dont l'incertitude de mesure sera détaillée dans la suite de ce chapitre. Enfin nous avons ajouté sur notre robot une plaque de 38.5 cm de coté, qui sera détectée depuis un laser Hokuyo 30 m positionné avec précision dans l'environnement (par triangulation avec l'angle du couloir). Ce dispositif est complété par un compas logiciel, dont nous détaillerons le fonctionnement par la suite, obtenu à partir des relevés des lasers embarqués, qui nous permet de limiter l'estimation de l'erreur de cap du robot.



a) Schéma de principe de la caractérisation dynamique des marqueurs Aruco

b) Caractérisation dynamique Aruco : Vue d'ensemble de la zone d'expérimentation

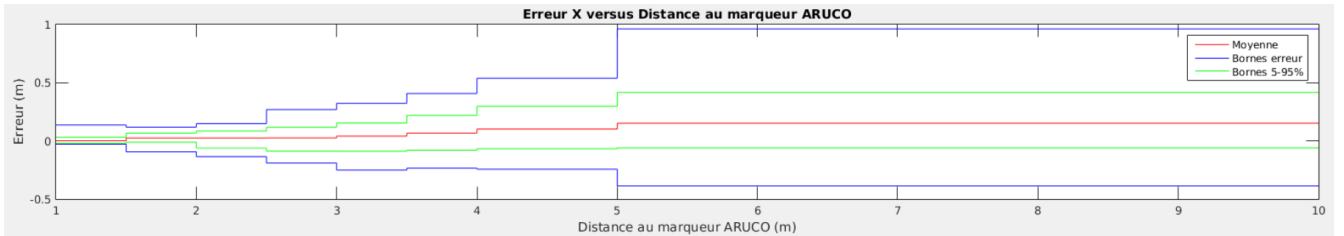
FIGURE 4.23 – Caractérisation dynamique des marqueurs Aruco sur la Kinect v.1

Ainsi ce dispositif permet d'obtenir plusieurs réalités terrain qui sont synchronisées dans le temps et fusionnées entre elles pour obtenir la meilleure précision possible. Cette réalité terrain, comparée avec les relevés Aruco obtenus depuis la Kinect v.1, nous permet alors d'obtenir les courbes d'erreurs de la Figure 4.24.

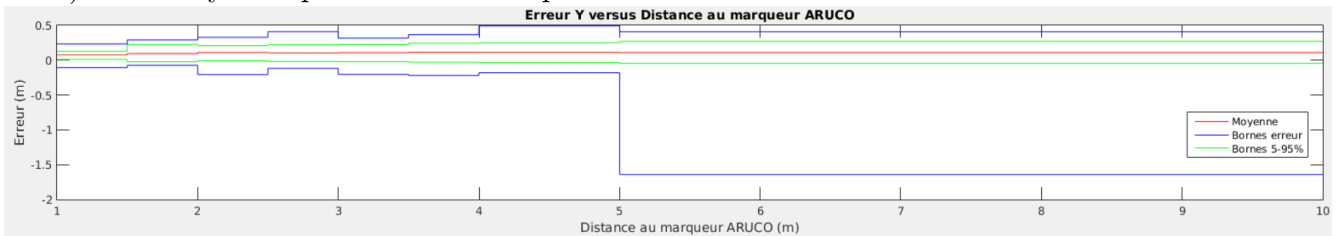
Selon X l'erreur, non centrée lorsque le marqueur est éloigné, qui est déjà de  $[-0.07, +0.07]$  à 1 m, se détériore régulièrement pour dépasser les 0.5 m au delà de 4 m. Mais globalement les résultats semblent plutôt meilleurs que statiquement même si, bien évidemment l'erreur observée reste importante. Selon Y, là encore, on obtient des résultats pratiquement centrés et acceptables jusqu'à 3 m. Au delà l'erreur s'accroît nettement. Enfin c'est selon l'orientation que le mouvement du robot détériore drastiquement l'erreur puisque cette dernière, bien qu'étant centrée explose dès 1.5 m pour atteindre près de 1 rad.

Les marqueurs Aruco utilisés avec la Kinect v.1 se révèlent relativement précis selon Y. En revanche selon X, leur précision se détériore régulièrement avec la distance de lecture. En ce qui concerne l'orientation ils sont pratiquement inutilisables en mouvement tant l'erreur devient rapidement importante. Ces résultats décevants sont en partie dus à la présence de mesures aberrantes. En effet si l'on exclue 5% des valeurs les plus extrêmes (voir courbes vertes Figure 4.24) on observe une forte diminution de l'erreur d'orientation en dessous d'une distance de détection de 3 m. D'autre part nos mesures intègrent l'ensemble des mesures effectuées pour

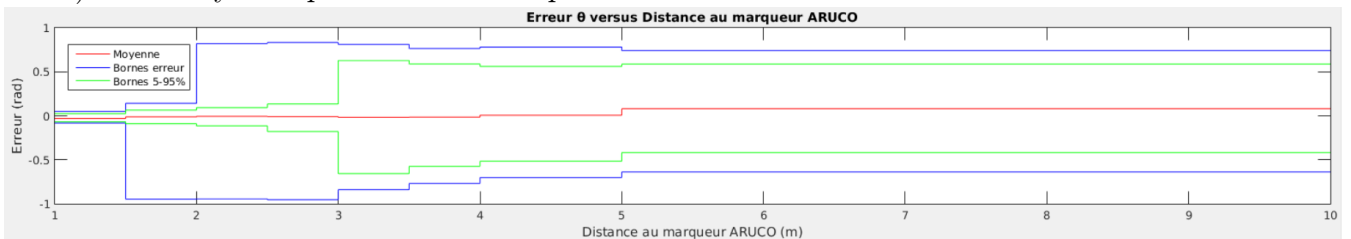
toutes les vitesses expérimentales. En établissant des relevés pour chacune de celles-ci on aurait observé la dégradation de l'erreur de mesures avec la vitesse du robot. Dans notre cas on retient le cas le plus défavorable. La relocalisation Kinect/Aruco nous permettra cependant de réduire l'incertitude de localisation en présence d'une dérive odométrique trop importante. Cependant, nous ne pourrions nous contenter d'une précision d'orientation aussi médiocre. C'est pourquoi, dans la suite du document nous introduirons un compas logiciel qui, à partir des données laser nous permettra, lorsque cela est possible de limiter l'incertitude d'orientation du robot.



a) Erreur dynamique en X des marqueurs Aruco



b) Erreur dynamique en Y des marqueurs Aruco



c) Erreur dynamique en  $\theta$  des marqueurs Aruco

FIGURE 4.24 – Caractérisation dynamique des marqueurs Aruco sur la Kinect v.1

Cependant les caractérisations statiques et dynamiques que nous venons de réaliser supposent que la Kinect est toujours capable de détecter un QR-Code lors de son déplacement. C'est une hypothèse forte à laquelle nous allons maintenant nous intéresser.

**C Influence du champ de vision** Les amers Aruco de QRC1 et QRC2 ont été disséminés dans l'environnement en utilisant la configuration de celui-ci (angles de couloir, piliers saillants, etc.) de façon à ce qu'ils soient positionnés préférentiellement perpendiculairement à la trajectoire suivie par le robot. Cependant, compte tenu de la distance séparant ces marqueurs, de la largeur du couloir, et du cap du robot, rien n'assure qu'il soit toujours possible que la Kinect 1 soit en capacité de détecter l'un d'entre eux pour recalibrer la valeur de la pose.

Arrêtons-nous dans un premier temps sur l'histogramme de la Figure 4.25 qui présente globalement, sur l'ensemble de la mission, le décompte des distances auxquelles les amers de QRC1 ont été détectés.

On peut en tirer plusieurs informations intéressantes :

- Globalement les QR-Codes sont détectés d'environ 1m à plus de 10m.

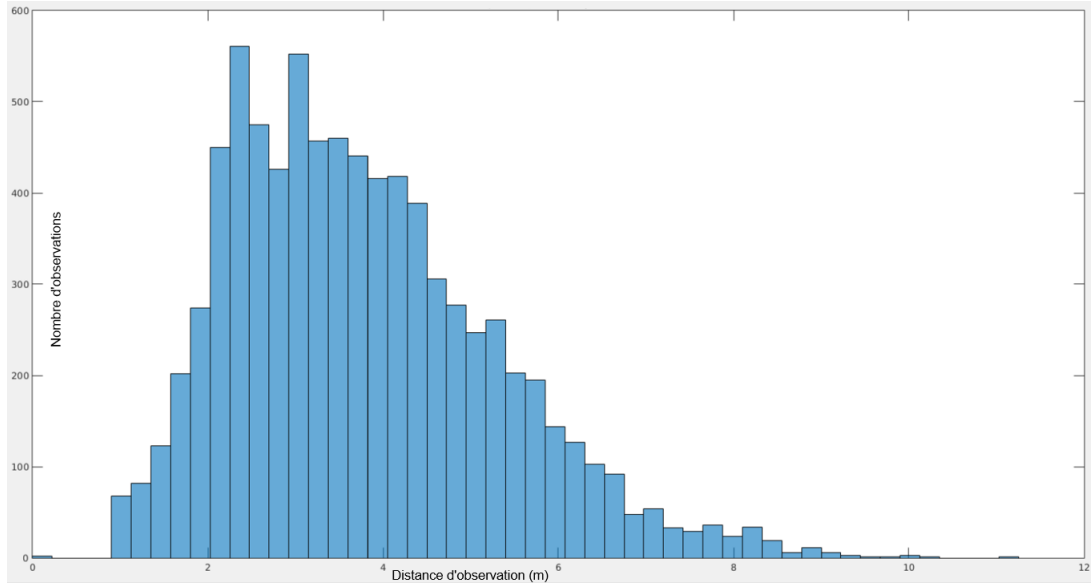


FIGURE 4.25 – Décompte de la distance de recalage au cours d’une mission

- C’est à une distance comprise entre 2m et 4.8m environ qu’ils sont le plus fréquemment détectés
- En dessous de 2m de distance de détection on note une décroissance significative du nombre de détections.

La difficulté de détection pour des distances proches ne serait pas pénalisante si l’on était certain de détecter au moins une fois chacun des QR-Codes. Ce n’est malheureusement pas le cas et une analyse détaillée de la détection de chaque amer démontre que certains ne sont jamais observés.

Pour comprendre l’origine de ce phénomène il est possible de tracer le graphique de la Figure 4.26 où sont représentés les éléments suivants :

- $D1 = 0.185\text{m}$  représente la distance entre l’origine du repère Robot et le référentiel Kinect
- $\theta_{ouv}$  représente le demi-angle d’ouverture de la Kinect qui est de  $29.5^\circ$  soit  $0.51\text{ rd}$ .
- $\theta_{cap}$  représente l’incertitude de cap du robot.
- Une ligne horizontale représente les positions extrémales où l’on souhaite pouvoir détecter un QR-Code, symbolisé ici par des rectangles bleus et rouges, placés sur un plan perpendiculaire au déplacement du robot à une distance  $Y_{QR} = \text{cte}$ .

On constate donc que l’on peut définir 2 cônes correspondant aux angles d’ouverture de la Kinect dont l’origine se trouve sur 1 cercle de rayon  $18.5\text{ cm}$ , coupé par des angles de  $\pm\theta_{ouv}$ . Pour une incertitude de cap donnée de  $2.\theta_{cap}$ , ces 2 cônes définissent alors 2 zones. Une zone orange tout d’abord où les 2 cônes ne s’intersectent pas. Il est alors **possible** de détecter un QR-Code placé à l’extérieur de la zone orange à  $Y_{QR}$  s’il se trouve à une distance minimale  $X_D$ . Ensuite, une zone verte où les 2 cônes s’intersectent à partir d’une abscisse  $X_F$ . On peut **garantir** de détecter un QR-Code placé à  $Y_{QR}$  à condition qu’il se trouve à l’intérieur de la zone verte à une distance minimale  $X_G > X_D$ .

On peut déterminer les formules analytiques suivantes (Équations 4.18, 4.19 et 4.20) liant  $X_D$ ,  $X_F$  et  $X_G$  à  $D1$ ,  $Y_{QR}$ ,  $\theta_{ouv}$  et  $\theta_{cap}$  avec  $Y_{cap} = D1.\sin\theta_{cap}$ .

$$X_D = \frac{(Y_{QR} - Y_{cap}) + D1.(\cos\theta_{cap}).\tan(\theta_{cap} + \theta_{ouv})}{\tan(\theta_{cap} + \theta_{ouv})} \quad (4.18)$$

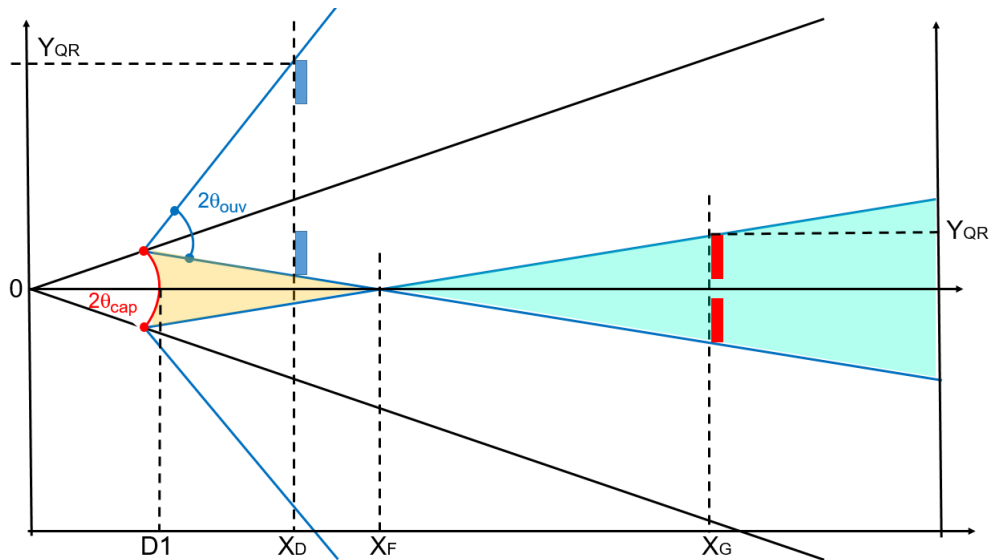


FIGURE 4.26 – Détection d'un QR-Code à l'aide d'une Kinect

$$X_F = \frac{Y_{cap} + D1 \cdot (\cos \theta_{cap}) \cdot \tan(\theta_{ouv} - \theta_{cap})}{\tan(\theta_{ouv} - \theta_{cap})} \quad (4.19)$$

$$X_G = \frac{Y_{QR} + X_F \cdot \tan(\theta_{ouv} - \theta_{cap})}{\tan(\theta_{ouv} - \theta_{cap})} \quad (4.20)$$

Ces dernières permettent de tracer l'abaque de la Figure 4.27 représentant l'évolution de  $X_D$  et  $X_G$  en fonction de  $\theta_{cap}$  pour  $Y_{QR} = 1\text{m}$ . Les lignes verticales noires représentent les erreurs de cap  $\theta_{cap}$  correspondant à la demi-amplitude de nos intervalles d'erreur d'orientation pour la Kinect 1.

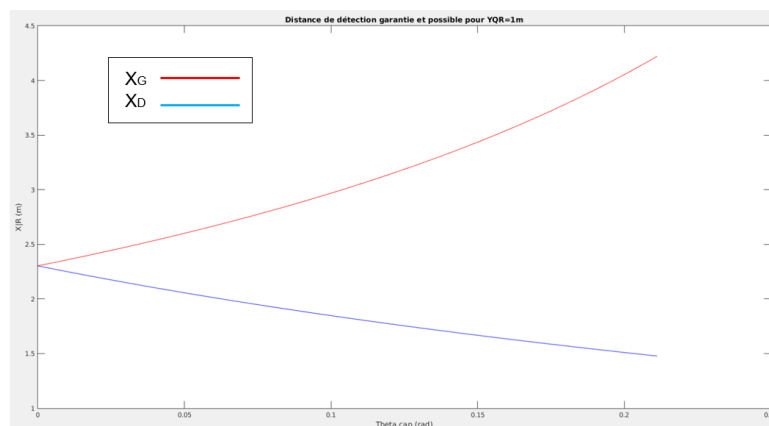


FIGURE 4.27 – Distance minimale possible et garantie d'un QR-Code excentré

Le raisonnement que nous venons d'effectuer démontre donc qu'un QR-Code excentré par rapport au sens de déplacement du robot doit se trouver à une distance minimale  $X_D$  de celui-ci pour pouvoir être détecté et à une distance supérieure  $X_G$  pour être assuré de le voir. Il n'est donc absolument pas certain de pouvoir repérer un amer excentré lorsqu'on s'en rapproche alors que l'incertitude de pose associée aurait été plus faible. On constate par ailleurs que la distance entre  $X_D$  et  $X_G$  croît significativement lorsque  $\theta_{cap}$  augmente. En conséquence, dans la suite de ce document, nous supposons donc que la distance de détection des QR-Codes est d'au

moins 2 m ( $>X_D$  mais  $<<X_G$ ). Ce choix semble raisonnable (cf Figure 4.25) mais il va nécessairement impacter la qualité garantie de nos recalages sans pour autant en assurer la détection.

Après avoir détaillé les performances de localisation de la Kinect 1 permettant d’analyser les marqueurs placés verticalement, passons maintenant à l’étude de la Kinect 2 dirigée vers les QR-Codes horizontaux placés au plafond.

#### 4.4.4.7 Kinect 2 et marqueurs Aruco

La Kinect v 1.0 est affectée d’un ensemble de limitations connues. Les données RGB sont de mauvaise qualité et fortement bruitées. Elle présente une répétabilité limitée de ses mesures [177]. C’est pourquoi une nouvelle version, la Kinect v. 2 (Figure 4.28), a été proposée en 2014. Elle fait appel à une caméra RGB 1920 x 1080 pixels, de meilleure définition, fonctionnant à la fréquence maximale de 30 Hz. La caméra infra-rouge voit aussi sa résolution augmenter à 512 x 424 pixels permettant de détecter des objets plus petits, grâce à une technologie basée sur la mesure d’un temps de vol (*Time of Flight*). Enfin ses champs d’ouverture sont agrandis. En particulier l’ouverture horizontale est portée à 70°. On peut cependant noter la suppression de la possibilité de contrôler le tilt qui existait sur la version 1.



FIGURE 4.28 – Le capteur Kinect v 2.0 à temps de vol

La caméra Kinect 2 que nous avons déjà utilisée dans le plan horizontal pour étudier l’incertitude des mesures de la Kinect 1 sera mise en œuvre pour analyser les marqueurs Aruco placés au plafond. L’intérêt de ce positionnement est que nous sommes en capacité de visualiser le couloir dans toute sa largeur et donc que, contrairement à ce que nous venons de démontrer avec la Kinect 1, nous pouvons garantir que nous détecterons les QR-codes du plafond dès que nous en serons assez proche. Nous allons donc maintenant caractériser ses performances dans le plan vertical.

**A Caractérisation dynamique** Nous nous affranchirons ici de la caractérisation statique de la Kinect 2 puisque, comme nous l’avons constaté pour la Kinect 1, les performances de mesure sont logiquement dégradées en mouvement. Ce travail a été réalisé dans la même zone d’expérimentation que pour la Kinect 1. Des marqueurs ont été positionnés au plafond tous les 1.5 m à une hauteur constante de 2.2 m. Les expériences ont été effectuées à une vitesse constante unique de 0.4 m/s pour un robot se déplaçant en ligne droite en boucle ouverte. La Kinect 2 ne peut plus être utilisée pour obtenir la réalité terrain de l’axe  $Y_M$ . Le laser 30 m est toujours employé pour déduire l’estimation en  $X_M$  de la position du robot. Celle de  $Y_M$  est obtenue à partir des informations laser en mettant en œuvre un algorithme de type RANSAC [81] (RANdom SAmple Concensus) pour retrouver la distance au mur droit du couloir. Enfin, comme pour la Kinect 1, l’orientation en cap est calculée en faisant appel à un compas logiciel qui sera présenté dans le prochain chapitre.

Le calcul de la localisation du robot à l'aide de la Kinect 2 faisant appel à des amers ARUCO placés au plafond induit une prise en compte tridimensionnelle de la position du QR-Code. Notre réalité terrain ne permettant d'observer que les erreurs de pose  $(x, y, \Theta)$ , nous ne pouvons identifier les erreurs affectant les dimensions  $z, \Phi$  et  $\Psi$  de l'espace 3D. En conséquence nous ne nous intéresserons qu'aux erreurs de pose et nous supposons que l'influence des erreurs impactant les autres dimensions a une influence négligeable sur le modèle expérimental de l'incertitude de pose de la Kinect 2 que nous déduirons de nos expériences.

La Figure 4.29 indique la distance en  $X_R$  à laquelle les QR-Codes placés au plafond sont détectés. On peut constater que ces derniers sont identifiés de façon certaine lorsqu'ils sont positionnés entre 0 et 40 cm à l'avant du robot. Il est parfois possible de les repérer vers l'arrière lorsqu'ils ne sont pas masqués par les 2 capteurs laser placés derrière et au-dessus de la Kinect 2.

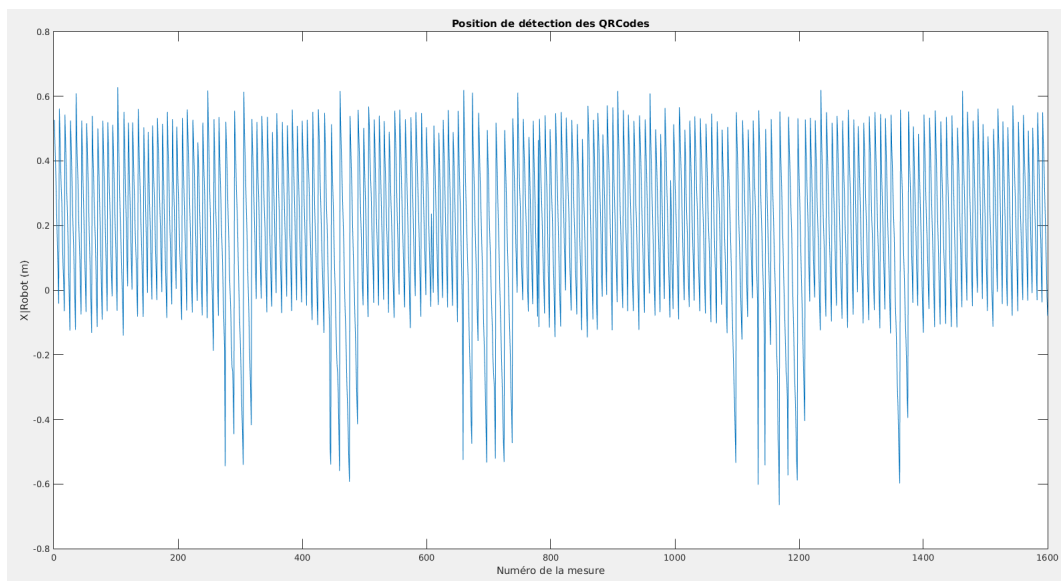


FIGURE 4.29 – Distance de détection de la Kinect 2 en dynamique

La Figure 4.30 qui représente les erreurs mesurées en  $(X,Y)$  montre une faible variabilité de ces dernières et permet de conclure sur les incertitudes en  $X$  et  $Y$ .

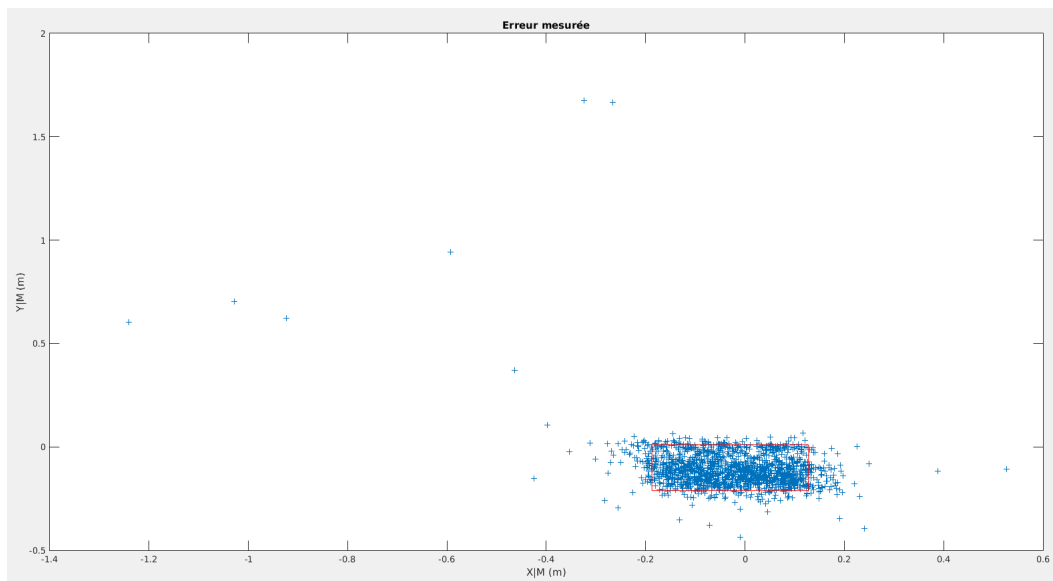


FIGURE 4.30 – Caractérisation dynamique de la Kinect 2



Pour plus de lisibilité, la Figure 4.31 présente l’histogramme des erreurs observées en  $\Theta$ . On constate que l’amplitude des erreurs de cap reste très faible avec un écart type réduit ( $\sigma = 0.0281rd$ ).

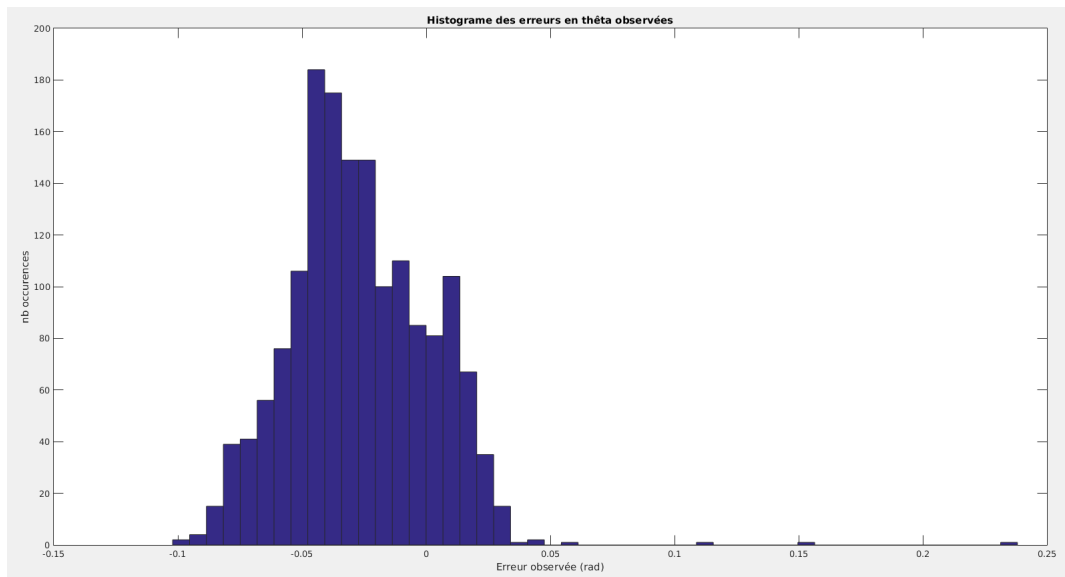


FIGURE 4.31 – Histogramme de répartition des erreurs en Théta de la Kinect 2

Si l’on exclut les 5% d’erreurs extrémales des relevés de mesures enregistrés on obtient les pavés d’incertitudes de la Table suivante.

Distances au QR-Codes (m)	Incertitude X (m)	Incertitude Y (m)	Incertitude d’orientation (rad)
[0, 0.4]	[-0.18, 0.12]	[-0.21, 0.01]	[-0.071, 0.017]

Incertitude 7 : Pavés d’incertitude Kinect2/Aruco en dynamique

On peut donc constater que, malgré une plage de détection seulement limitée à 40 cm, les performances affichées par la Kinect 2 permettent de réaliser des recalages de bien meilleure qualité que ceux de la Kinect 1 tout en garantissant de pouvoir observer les QR-Codes du plafond.

L’ensemble des capteurs de notre plateforme ayant été présenté et si nécessaire caractérisé, nous allons maintenant rapidement décrire les moteurs qui permettent au robot de se déplacer.

#### 4.4.5 Actionneurs

Notre plateforme robotique est dotée de 2 roues actionnées par des moteurs à courant continu 12 V réversibles Pittman 12V GM 9236 (Figure 4.32). Leur vitesse de rotation est contrôlée à l’aide de PWM de fréquence de hachage de 20 KHz.

Il est aussi important de noter que la documentation Pioneer stipule que les vitesses de rotation des roues ne peuvent être contrôlées qu’à 0.2 rad/s près. Par ailleurs, évidemment, les changements de vitesse de rotation ne s’effectuent pas instantanément, dans les limites fixées, mais, à bas niveau, selon une commande trapézoïdale en vitesse (Figure 4.33), contrôlée par un système PID (*Proportional-Integral-Derivative*) permettant d’ajuster la fréquence des PWM et donc la vitesse de rotation des roues toutes les 5 ms. Les gains du PID peuvent être modifiés pour maîtriser au mieux la réponse du système aux demandes de changement de vitesse.



FIGURE 4.32 – Pioneer P3-Dx : Moteur Pittman 12V 9236

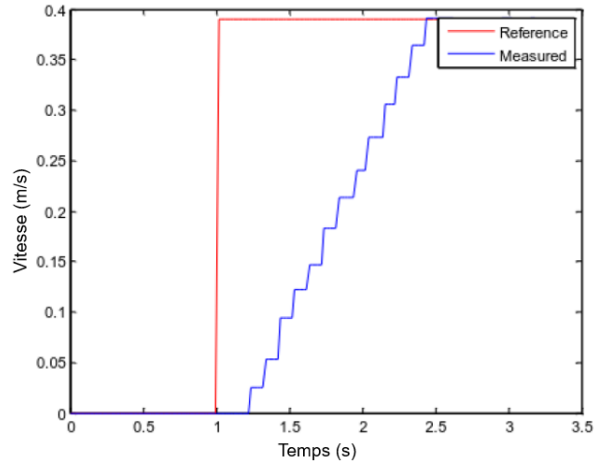


FIGURE 4.33 – Rampe de commande du robot Pioneer

#### 4.4.6 Conclusion

Nous venons de balayer l'ensemble de la vue matérielle de notre plateforme robotique en détaillant principalement son organisation spatiale ainsi que l'ensemble de ses capteurs. Au fil de notre discours, nous avons précisé les pavés d'incertitudes induits pour chacun de ces éléments. Nous allons maintenant nous intéresser à la dimension logicielle de notre plateforme et aux incertitudes de localisation pouvant être induites tant par la prise en différé des données que par la durée de leur traitement.

### 4.5 La plateforme robotique : Vue logicielle

La dimension logicielle est bien évidemment très importante pour le calcul de l'incertitude de localisation. En effet la mise en œuvre de la boucle de "Navigation - Guidage - Contrôle" nécessite d'être capable de recueillir les informations capteurs, de les mettre en forme (élimination des données aberrantes, filtrage, changement de référentiel, etc..) de façon à ce que ces dernières puissent être utilisées pour déterminer la position supposée du robot et permettre le calcul de la commande pour qu'il puisse rejoindre son prochain point de passage selon une stratégie choisie. Évidemment les fréquences d'acquisition des données capteurs varient en fonction de la nature de ces derniers, les algorithmes de localisation doivent être déployés avant que la commande puisse être déterminée et appliquée. Il faut donc mettre en œuvre une architecture logicielle à même de gérer de concert, et avec le minimum de dérive, tant l'exécution séquentielle de la boucle de contrôle commande, que l'asynchronisme des lectures capteurs. La gestion des échanges de données entre ces différents composants logiciels et matériels se trouve facilitée lorsque l'on fait appel à un middleware.

### 4.5.1 Le middleware : CONTRACT

CONTRACT (*Control Architecture Creation Technology*) [181] [182] est un middleware développé au LIRMM depuis une quinzaine d'années qui permet la construction d'architectures logicielles temps-réel composées de modules décisionnels (superviseurs) et exécutifs pouvant être associés en schémas. Ces principaux points forts sont de permettre un contrôle très fin de la couche exécutive en assurant un déterminisme d'exécution tout en permettant une reconfiguration des schémas d'exécution à la volée.

Un *module* CONTRACT est composé logiquement d'une machine à états permettant de gérer son comportement (création, destruction, liens externes -réception/production de données, d'événements-, code interne), ce code interne décrivant la fonctionnalité du module considéré. Il peut être synchrone (exécution périodique), ou asynchrone (exécution sur événement). Outre les ports d'entrée et de sortie de données qui permettent aux modules de communiquer, il dispose aussi de ports spécifiques associés à la gestion/production d'événements, où permettant de contrôler son activité (démarrage, arrêt, configuration).

Un schéma correspond à un ensemble de modules périodiques interconnectés, exécutés selon une période fixe, dont l'exécution peut respecter des règles de priorité. Chaque schéma s'exécute indépendamment, de façon asynchrone, selon la fréquence qui lui est associée.

L'architecture logicielle se décompose en 2 couches distinctes et complémentaires (Figure 4.34). D'une part une *couche décisionnelle* fonctionnant de façon asynchrone et donc totalement événementielle. Elle est composée de superviseurs organisés hiérarchiquement qui assurent la prise de décision contextuelle en fonction des événements reçus. En fonction du contexte, cette couche détermine donc quels sont les schémas devant s'exécuter et si nécessaire leur paramétrage. D'autre part la *couche exécutive* composée des modules (pouvant être regroupés en schémas) mettant en œuvre les fonctionnalités robotiques. On distingue le domaine synchrone où les modules sont exécutés périodiquement sous le contrôle d'un *ordonnanceur*, du domaine asynchrone dont l'exécution ne dépend que de la couche décisionnelle.

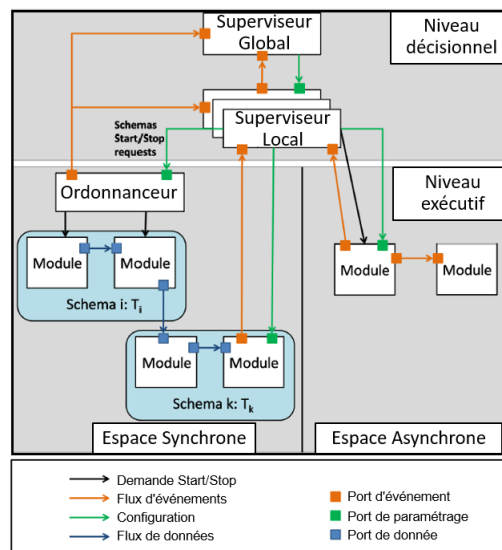


FIGURE 4.34 – CONTRACT : Schéma de principe de l'architecture logicielle

Une nouvelle version de l'architecture CONTRACT encore plus performante est en cours de développement. Elle vise à assurer une meilleure gestion de la dynamique de composition des fonctionnalités, de déploiement (appariement fonction/processus), l'intégration d'une gestion

multi-coeurs complète permettant de créer des pipelines synchrones et asynchrones concurrents, et enfin la programmation des couches de supervision via une API (*Application Programming Interface*) plutôt qu'un DSL (*Domain Specific Language*) pour en faciliter la maintenance.

Les grands principes du middleware CONTRACT étant énoncés nous allons maintenant présenter les modules déployés dans le cadre de notre mission en apportant une attention particulière à leur fréquence d'exécution et aux pipelines exécutifs qui vont nécessairement induire des délais de traitement affectant la précision de localisation.

## 4.5.2 Architecture logicielle : Modules, discrétisation et incertitude de localisation

Dans cette partie nous allons commencer par décrire les modules mis en œuvre pour exécuter la mission de patrouille qui nous intéresse en précisant leur durée maximale et/ou fréquence d'exécution. Dans un second temps, nous nous intéresserons aux délais induits par la discrétisation du traitement des informations tout au long de la chaîne de contrôle-commande, et nous estimerons l'impact de cette latence sur la précision de localisation.

Avant d'entamer la description des modules de l'architecture il est important de préciser que celle-ci est mise en œuvre sur le PC-embarqué de la plateforme robotique. Ce dernier communique avec le contrôleur bas-niveau du Pioneer par le biais d'une liaison série USB et d'un mécanisme de type client-serveur selon un protocole d'échange propriétaire.

### 4.5.2.1 Les modules

Afin de structurer cette description nous allons tout d'abord présenter les modules en liaison directe avec les capteurs et actionneurs avant de nous intéresser à ceux mettant en œuvre des fonctionnalités robotiques. Nous terminerons par ceux plus spécifiquement liés à la gestion de la mission et à la méthodologie PANORAMA.

Nom	Fonction	Durée (ms)	Période (ms)
LAS1	Valeurs brutes laser frontal	-	100
LAS2	Valeurs brutes laser arrière	-	100
KINECT1	Image Kinect 1 brute	-	33
KINECT2	Image Kinect 2 brute	-	33
P3D	Contrôle robot et valeurs brutes odométrie - sonars	10	100

TABLE 4.4 – Modules de liaison avec la partie matérielle

La Table 4.4 présente les modules permettant de recueillir les données capteurs de notre plateforme ainsi que le contrôle de la vitesse des roues du robot. On y trouve évidemment les 2 modules LAS1 et LAS2 qui permettent de récupérer les données laser brutes à la fréquence de 10 Hz imposée par le constructeur. KINECT1 et KINECT2 sont les modules en charge de la récupération des données des caméras Kinect à la fréquence théorique de 30 Hz. On y trouve aussi le module P3D qui joue un rôle central dans l'architecture. En effet il permet à la fois de contrôler le robot (vitesse roue, fréquence ultra-son, gain PID, etc.), mais aussi de récupérer lors de chaque sollicitation, les valeurs mémorisées de l'odométrie, des banques d'ultrasons et

des capteurs de choc, que le contrôleur bas niveau renvoie. Il est sollicité périodiquement à la fréquence de la boucle de contrôle-commande qui a été fixée à 10 Hz.

La Table 4.5 énumère les modules nécessaires à la localisation du robot. P3D est encore présent puisqu'il donne directement la position du robot dans  $\mathcal{R}_R$ . Les algorithmes auxquels font appel ces modules seront détaillés dans le prochain chapitre.

Nom	Fonction	Durée (ms)	Période (ms)
P3D	Odométrie - Navigation à l'estime	10	100
KINODO	Suivi de posture Odométrie et recalage Kinect 1	15	33
KIN2ODO	Suivi de posture Odométrie et recalage Kinect 2	15	33
COMPAS	Estimation orientation	5	100
LZA	Localisation absolue avec grille	60	100

TABLE 4.5 – Architecture : Modules de LOCALISATION

On trouve dans la Table 4.6 les modules pouvant être employés pour les déplacements du robot. On utilise une méthode de suivi de chemin (SCH) à garantie de convergence, utilisable sur l'ensemble de la carte, où le robot va chercher à rejoindre asymptotiquement le chemin puis à suivre un lièvre virtuel s'y déplaçant. Nous avons donc à notre disposition jusqu'à 5 alternatives d'implémentation pour déplacer le robot. Tout déplacement s'effectue en sécurité en faisant appel à une méthode d'évitement d'obstacle dite SMZ (Safe Maneuvering Zone)[143]. En cas d'impact d'obstacle trop proche, la trajectoire est localement modifiée et le robot suit un nouveau lièvre placé sur un cercle d'évitement jusqu'à ce qu'il rejoigne à nouveau la trajectoire initiale.

Nom	Fonction	Durée (ms)	Période (ms)
SCH_US	Suivi de chemin avec les US	2	100
SCH_LAS1	Suivi de chemin avec laser frontal	2	100
SCH_LAS1US	Suivi de chemin avec laser frontal et US	2	100
SCH_LAS12	Suivi de chemin avec les 2 lasers	2	100
SCH_LAS12US	Suivi de chemin avec les 2 lasers et les US	2	100

TABLE 4.6 – Architecture : Modules de DÉPLACEMENT

La rotation sur place du robot (ROTODO) pour analyser l'état d'une vanne (Table 4.7) est effectuée uniquement en se fiant à l'odométrie. On réalise la rotation pour aligner le robot à une orientation donnée.

Nom	Fonction	Durée (ms)	Période (ms)
ROTODO	Rotation à l'odométrie	10	100

TABLE 4.7 – Architecture : Modules de ROTATION

Le module d'analyse d'image ANIM (Table 4.8) repère dans les données Kinect reçues la position de la vanne rouge et en fonction de son orientation (avec certains seuils) par rapport à l'horizontale, conclut sur son état ouvert ou fermé.

Nom	Fonction	Durée (ms)	Période (ms)
ANIM	Détermination état d'une vanne	20	-

TABLE 4.8 – Architecture : Module D'ANalyse d'Image

Pour terminer la Table 4.9 rassemble les modules utilisés pour mettre en œuvre la méthodologie PANORAMA. Le module WATT permet de suivre la consommation énergétique du robot (puissance, courant, tension, temps écoulé). RELAIS sert à contrôler les relais pour mettre sous tension les capteurs connectés et gérer ainsi l'énergie consommée. OBSERV est un module très important car il permet de suivre le déroulement de la mission et met à jour les marges de performance, la position linéaire du robot au sein de sa mission, ainsi que la liste des ressources opérationnelles. Enfin le module central de la méthodologie est PANORAMA puisqu'il permet la mise en œuvre de la méthodologie proposée. Il construit le plan de mission initial en fonction des contraintes imposées. C'est lui qui sélectionne, activité par activité, les schémas à utiliser et leur paramétrage. Ce travail est réalisé hors ligne. De façon asynchrone, il est à nouveau sollicité depuis OBSERV lorsqu'une activité est achevée, qu'une marge devient négative, ou qu'une ressource est défaillante. Dans le cadre de ce travail la construction du SMD a été réalisée par nos soins, et seule la sollicitation asynchrone d'OBSERV lors du changement d'activité a été mise en œuvre.

Nom	Fonction	Durée (ms)	Période (ms)
WATT	Gestion du wattmètre embarqué	30	100
RELAIS	Gestion des relais	1	-
OBSERV	Suivi de mission	2	100
PANORAMA	Gestion de la mission	1	-

TABLE 4.9 – PANORAMA : Modules de gestion de la mission

Tous les modules que nous venons de balayer constituent, lorsqu'ils sont impliqués dans les tâches robotiques courantes, autant de processus légers ou *threads* gérés par l'ordonnanceur de CONTRACT. Nous allons maintenant nous attarder sur le schéma d'exécution NGC de la chaîne de Navigation-Guidage-Contrôle qui permet au robot de se déplacer et va être à l'origine de l'incertitude de Traitement Logicielle impactant la localisation.

#### 4.5.2.2 Incertitude de Traitement Logiciel

Dans ce paragraphe nous allons commencer à expliciter le déroulement du schéma NGC avant d'en déduire l'erreur de localisation induite en raison du traitement séquentiel et périodique qui entraîne une latence dans le traitement des informations capteurs.

Les schémas NGC présentés en Figure 4.35 représentent l'enchaînement des traitements logiciels mis en œuvre dans le cadre d'une navigation exclusivement à l'estime, ou faisant appel à des capteurs extéroceptifs pour se localiser.

- Dans le cas d'une localisation à l'estime, toutes les 100 ms, les informations odométriques sont récupérées en début de boucle avant d'être mises en forme pour obtenir la localisation du robot. Celle-ci est alors utilisée dans le module de guidage, avec les informations de navigation (chemin, prochain point de passage) pour calculer les vitesses des roues, qui sont alors appliquées grâce au module de contrôle.

- Dans le cas d'une localisation pouvant faire appel aux informations interoceptives et extéroceptives, la boucle NGC utilise des données brutes de capteurs générées de façon asynchrone, à une fréquence imposée par la boucle elle-même (odométrie - 100 ms), ou par les capteurs (Kinect - 30 ms, Laser - 100 ms). Celles-ci sont tout d'abord mises en forme puis utilisées par le module de Localisation choisi pour en déduire la localisation du robot. Cette dernière permet alors, comme dans le cas précédent, au travers des modules de Guidage et de Contrôle, de maîtriser la trajectoire du robot.

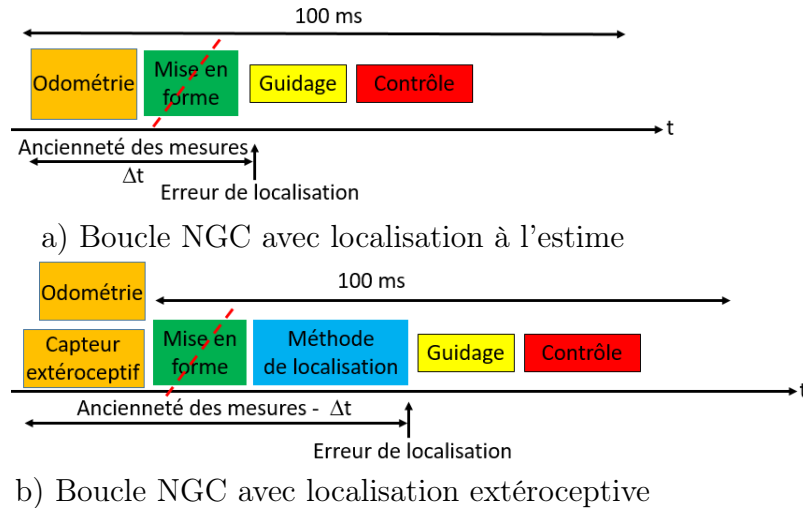


FIGURE 4.35 – Schémas d'exécution des boucles de Navigation-Guidage-Contrôle

Rappelons ici que l'incertitude de localisation est considérée au moment de l'utilisation de la pose estimée du robot par le module de guidage (Figure 4.35). On constate donc que celle-ci est calculée avec des informations capteurs plus ou moins anciennes alors que le robot s'est encore déplacé depuis leur obtention. Dans le cas de la localisation à l'estime il est possible de négliger la durée de la mise en forme des données odométriques et l'on peut donc considérer que l'information capteur reflète alors la position réelle du robot, si l'on oublie la dérive odométrique évidemment. Les délais induits par l'architecture de contrôle n'ont alors pas d'impact sur l'erreur de localisation. Il n'en est pas de même lorsque l'on fait appel à des capteurs extéroceptifs. C'est sur ce point que nous allons concentrer notre attention.

Puisque la méthode de localisation utilisée dans la boucle NGC utilise des informations capteurs anciennes nous allons quantifier le pavé d'incertitude induit par la manipulation des données au sein de l'architecture de contrôle estimant, dans le cas le plus défavorable, le chemin parcouru et la variation de cap depuis leur acquisition jusqu'à la mise à disposition de la pose au niveau du module de guidage. Cette durée est représentée par  $\Delta t$  dans la Figure 4.35.  $\Delta t$  correspond donc à la somme de la période capteur la plus grande, du temps de mise en forme (négligeable), et de la durée de calcul de la méthode de localisation.

Pour y parvenir nous allons nous appuyer sur la dernière commande appliquée et supposer que la trajectoire visée a bien été suivie. Il est alors possible de calculer le chemin parcouru le long de l'abscisse curviligne de la trajectoire suivie pendant  $\Delta t$  à partir de la connaissance de la vitesse de rotation des roues. En se référant à la Figure 4.36 il est possible de déduire classiquement [204] une estimation de l'erreur de pose selon  $X$ ,  $Y$  et  $\theta$  présentées en Équation 8 et calculées à partir d'une estimation odométrique du déplacement pendant  $\Delta t$ .

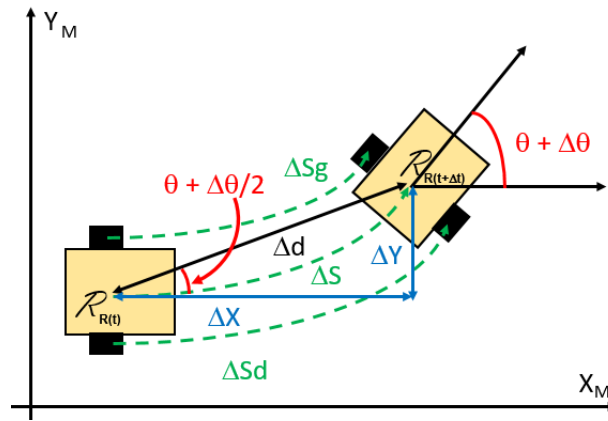


FIGURE 4.36 – Erreur architecture et modèle de déplacement du robot

$$\begin{aligned}\Delta X &= \Delta S \cdot \cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \Delta Y &= \Delta S \cdot \sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \Delta\theta &= \frac{\Delta S_d - \Delta S_g}{b}\end{aligned}$$

Incertitude 8 : Incertitudes de Traitement Logiciel

où :

- $b$  représente la distance entre les 2 roues du robot
- On suppose que  $\Delta d \sim \Delta S$
- $\Delta S = \frac{\Delta S_d + \Delta S_g}{2}$

Cette formulation peut engendrer une erreur de pose non négligeable lorsque la vitesse croît au moment de leur prise en compte par le guidage. Ajoutons cependant que cette erreur de traitement logiciel est prise en compte et compensée par le module de guidage pour le calcul de la commande à appliquer. C'est pour cela que nous ne l'avons pas considérée dans notre modèle d'incertitude de pose.

### 4.5.3 Conclusion

Le traitement logiciel des informations capteurs et leur manipulation au sein de la boucle de contrôle induit une latence conduisant à la détermination d'une pose du robot indépendamment du mouvement du robot. L'incertitude engendrée va donc principalement dépendre de la fréquence de mise à disposition des informations capteurs et de la complexité algorithmique associée à la méthode de localisation employée. Fort heureusement cette latence est compensée, à la fin de la boucle NGC, lors du calcul du guidage de la plateforme robotique.

## 4.6 Conclusion

Ce chapitre a permis de préciser, tant sur le plan matériel que logiciel, la valeur des pavés d'incertitude des principales sources d'erreurs qu'il nous est possible de déterminer objectivement sur notre plateforme. Nous sommes bien évidemment pleinement conscient que nombre d'entre-elles, bien qu'existantes, ont été négligées ou sont encore passées sous silence. Il a par ailleurs permis d'identifier les référentiels mis en jeu dans le calcul de la pose d'un robot ainsi que les équations matricielles les reliant.



Mais n'oublions pas que la méthodologie PANORAMA nécessite de disposer de modèles prédictifs permettant de prédire les performances qui devraient être observées tout au long d'une mission. Le prochain chapitre va donc s'appuyer sur l'ensemble des résultats que nous venons de présenter dans ce chapitre pour proposer un modèle intervalliste prédictif de l'erreur de pose pour les différentes méthodes de localisation mises en œuvre dans notre application.

# Chapitre 5

## PERFORMANCE DE LOCALISATION ET MODÈLES PRÉDICTIFS D'INCERTITUDE

Le déploiement de la démarche PANORAMA dont l'objectif est de permettre la mise en œuvre de missions robotisées autonomes à garantie de performance nécessite la définition de modèles prédictifs de performance. Le chapitre précédent nous a permis d'identifier, sur notre plateforme, tant matériellement que logiciellement, les principales composantes influençant l'incertitude de pose des méthodes de localisation.

Ce chapitre va nous permettre d'établir les modèles d'incertitude des méthodes de localisation que nous avons déployées et que nous espérons représentatives de celles pouvant être rencontrées en robotique. Nous allons donc mettre en œuvre une démarche méthodologique qui pourrait être utilisée pour d'autres approches de localisation.

Nous présenterons dans un premier temps le processus d'identification et de caractérisation des modèles prédictifs de localisation. Puis nous le projeterons sur les différentes méthodes déployées. Nous commencerons tout d'abord par la démarche classique de navigation proprioceptive à l'estime basée sur l'odométrie qui nous permettra d'établir un modèle analytique d'évolution de l'erreur de pose. Nous nous intéresserons ensuite à des méthodes faisant appel à un recalage extéroceptif et nous baserons nos modèles d'erreur sur la construction de modèles d'erreur ensemblistes prédictifs. Nous analyserons tout d'abord une méthode de recalage visuel d'amers artificiels faisant appel à une caméra, puis nous présenterons et analyserons une nouvelle approche de localisation absolue LZA, basée sur les apparences, et développée au sein de l'équipe EXPLORE. Dans ces deux cas nous chercherons à en améliorer les performances en les fusionnant avec des informations relevant d'autres modalités que celles déjà utilisées. Pour finir nous tenterons de porter un regard critique sur les résultats observés.

### 5.1 Performance de localisation : Démarche mise en œuvre

Quelle que soit la méthode de localisation étudiée, la démarche d'élaboration des modèles d'erreurs prédictifs de la pose va suivre la même séquence procédurale. Les phases qui la composent (Figure 5.1) peuvent être obligatoires (en vert), optionnelles (en orange) ou complémentaires (en bleu) en fonction des éléments à disposition et de la garantie obtenue.

La Phase 1 correspond à l'étape préliminaire de la démarche PANORAMA. Elle consiste à identifier les liens de causalité reliant les différents facteurs pouvant impacter un objectif

de performance et donc, dans notre cas, la précision de pose pour une méthode de localisation donnée. Ce travail préliminaire est loin d'être anodin car il permet de déterminer les indicateurs (objectif) et inducteurs de performance ainsi que leurs liens causaux. L'analyse de la nature des inducteurs permettra de mettre en évidence, s'ils existent, des leviers sur lesquels il est possible de jouer avant ou pendant la mission, pour impacter et donc piloter le niveau de performance observé.

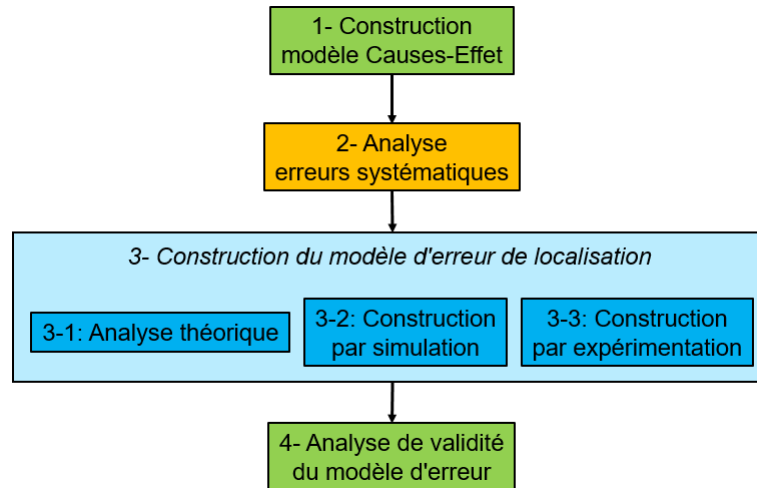


FIGURE 5.1 – Procédure de détermination d'un modèle d'erreur prédictif

Parmi les outils permettant d'explicitier ces liens de causalité tels que les diagrammes ou matrices d'interactions [45], nous nous sommes tournés vers le diagramme d'Ishikawa [111]. Dans le domaine de l'entreprise, ce dernier permet de traduire graphiquement et donc qualitativement, au travers d'une analyse 5-M (Matière, Matériel, Méthode, Milieu, Main d'œuvre), les causes (inducteurs) impactant un effet donné (indicateur de performance). Nous l'adapterons donc au contexte de notre étude.

La Phase 2 est optionnelle. En effet, si l'analyse du modèle causes-effets met en évidence des biais il est important de mettre en œuvre les procédures permettant de les identifier et de les compenser du mieux possible. Ce sera par exemple le cas dans notre étude pour les erreurs systématiques qui peuvent affecter le modèle de dérive odométrique.

Il existe ensuite différents chemins (Phase 3) pouvant conduire à l'élaboration d'un modèle prédictif au sein d'un environnement donné. Matérialisés en bleu dans la Figure 5.1 ils correspondent aux différentes alternatives. Ces dernières peuvent être complémentaires, possèdent chacune leurs propres limitations, et au moins l'une d'entre-elles doit être mise en œuvre pour obtenir un modèle prédictif.

- Lorsque cela est envisageable il est possible d'effectuer une analyse que nous qualifierons de théorique de la méthode de localisation. Celle-ci peut recouvrir différentes approches qui peuvent parfois être composées :
  - On peut tout d'abord construire un modèle théorique de l'évolution de l'incertitude de localisation (Phase 3.1). Comme pour tout modèle, ses limites seront liées aux hypothèses et aux simplifications retenues pour sa construction. Compte tenu de l'impact de l'environnement sur l'incertitude de localisation d'une méthode, pour construire un modèle prédictif, ce type d'approche se trouve limité aux méthodes proprioceptives. Une fois les équations établies, il doit impérativement être complété par une phase d'expérimentations pour permettre de quantifier la valeur des para-

mètres du modèle. C'est la démarche que nous suivrons pour construire le modèle prédictif odométrique.

- Il est aussi possible de s'appuyer sur un modèle couplant, par construction, les dimensions expérimentales et théoriques. C'est typiquement le cas de méthodes pour lesquelles les mesures affichent un comportement déterministe vis-à-vis de l'environnement, et où l'erreur de localisation peut-être reconstituée globalement en composant les pavés d'incertitude établis expérimentalement au moyen des équations de changement de référentiels. C'est la démarche que nous suivrons par exemple pour la méthode de recalage par détection d'amers artificiels.
- Il est aussi possible de mettre en œuvre des campagnes de simulation massives (Phase 3.2) où des scénarios différents sont joués. Pour chacun d'eux, une analyse du comportement de l'erreur de localisation peut être effectuée. La simulation présente de nombreux intérêts. Elle peut servir à évaluer les paramètres qui peuvent influencer les performances d'une méthode de localisation. Elle permet d'identifier les zones de l'environnement où la méthode étudiée peut, ou non, être employée. Elle permet de qualifier par zone, la garantie de localisation qu'une méthode peut procurer. Bien évidemment la qualité de ces résultats va dépendre de la qualité des modèles mis en œuvre pour simuler le comportement de la plateforme (modèles capteurs, robot, environnement, etc.) mais aussi de la masse de simulations effectuées. Ce type d'étude sera exploité dans le cas de l'analyse de la méthode de localisation absolue LZA.
- Enfin il est aussi envisageable de s'appuyer sur des relevés expérimentaux (Phase 3.3) pour construire les modèles prédictifs d'évolution de l'erreur de localisation. L'environnement doit donc être instrumenté de façon à permettre la reconstruction d'une réalité terrain indépendamment de la méthode de localisation employée. Plusieurs cas de figures peuvent se présenter :
  - Si aucune analyse méthodologique de la méthode n'a pu être effectuée, on obtient alors un modèle d'erreur purement expérimental de type "boite noire". Les résultats obtenus vont donc dépendre de l'environnement et des conditions expérimentales. Pour fiabiliser le modèle prédictif observé il faudra donc itérer les relevés expérimentaux sans pouvoir assurer une réelle garantie. En effet les conditions les plus défavorables n'auront peut-être pas été rencontrées. Il est cependant illusoire de procéder à cette analyse sur l'ensemble de l'environnement.
  - En revanche, si une analyse méthodologique permet de définir une localisation garantie, il est alors des plus pertinents de s'assurer qu'expérimentalement, sur un environnement représentatif, l'erreur de pose observée s'inscrit bien dans les limites prévues théoriquement.

Dans le cadre de ce document nous aurons systématiquement recours à cette phase expérimentale pour toutes les méthodes de localisation étudiées. Nous nous appuyerons sur la zone d'expérimentation instrumentée que nous avons déjà présentée dans le chapitre 4 pour caractériser la performance de localisation dynamique des marqueurs Aruco.

Pour terminer, au regard des résultats obtenus la Phase 4 finale permet de conclure sur le modèle prédictif construit et sur la garantie qu'il est à même ou non d'assurer. Si l'on ne dispose que de modèles expérimentaux la confiance de garantie doit être manipulée avec précaution.

De la même façon si l'on ne considère qu'une analyse faite par simulation, elle doit être analysée en fonction de la validité des hypothèses retenues et des modèles utilisés. Quoi qu'il en soit les relevés expérimentaux sont indispensables pour venir corroborer la validité des modèles construits théoriquement ou à partir de simulations. Toute incohérence viendrait immédiatement disqualifier le modèle prédictif proposé.

La démarche générale permettant l'élaboration des modèles prédictifs de localisation étant présentée nous allons maintenant la décliner sur les méthodes de localisation déployées dans le cadre de notre mission en précisant les phases de la démarche dont elles relèvent.

## 5.2 ODO : Localisation proprioceptive odométrique

Pour cette méthode proprioceptive de localisation nous commencerons par un bref état de l'art avant d'en préciser le modèle causes-effets pour terminer en identifiant les erreurs statiques et en établissant un modèle analytique de la dérive odométrique s'appuyant sur un ensemble de relevés expérimentaux.

### 5.2.1 État de l'art

L'approche odométrique reste très couramment utilisée pour estimer la pose d'un robot en raison de son coût (financier, énergétique, calculatoire) réduit, de sa fréquence d'échantillonnage élevée et de sa bonne précision sur des distances réduites. Pour en améliorer la précision les premiers travaux ont cherché à réduire l'incertitude de mesure de la vitesse de rotation des roues et par la même l'incertitude de pose [2] [131]. Ces dernières années de très nombreuses études ont cherché à fusionner les informations odométriques avec celles issues d'autres capteurs [215] [170] [138].

Pour notre part nous nous focaliserons ici sur les travaux développés principalement aux alentours des années 2000 pour caractériser l'erreur odométrique en tant que telle. Ceux-ci distinguent classiquement deux types d'erreurs odométriques :

- Les **erreurs systématiques** qui trouvent leurs origines dans l'imprécision des paramètres cinématiques du robot telles que des diamètres de roues inégaux, une mauvaise connaissance de l'empattement ou de la position du centre de masse. Leurs valeurs effectives peuvent en effet différer des paramètres constructeurs ou en fonction de la charge du robot. On peut trouver dans [51] une étude théorique très intéressante de l'impact de ces incertitudes paramétriques sur l'estimation de pose et de vitesse du robot. Ces erreurs, dont l'effet est prédominant sur une surface lisse [35], sont à l'origine d'un biais de mesure.
- Les **erreurs non systématiques** résultent des interactions du robot avec son environnement en présence d'une surface de déplacement inégale ou de glissements induits par des forces externes (choc), ou internes (rotation).

De nombreux travaux ont cherché à caractériser ces erreurs de façon à les compenser et par là même à améliorer la précision de l'odométrie. Ces techniques de calibration se décomposent en 3 catégories distinctes.

- D'une part celles où l'on fait suivre au robot des trajectoires prédéterminées pour mettre en évidence l'impact des sources d'erreurs. Les travaux de Borenstein [35] font références pour l'estimation des paramètres cinématiques. La méthode UMBmark proposée permet de compenser les erreurs systématiques en faisant effectuer au robot des trajectoires carrées. Dans [227] les auteurs s'intéressent à une trajectoire circulaire pour déterminer la matrice de covariance caractérisant la dimension stochastique des erreurs odométriques non systématiques. La position du robot dépendant non linéairement des changements en translation et en orientation, le problème est linéarisé en faisant appel à un développement de Taylor.
- D'autre part des approches permettant d'effectuer la calibration de l'odométrie, en suivant des trajectoires quelconques. Dans [61], les auteurs effectuent une analyse critique des

travaux antérieurs, proposent et mettent en œuvre un modèle analytique pouvant s’appliquer à une trajectoire quelconque. Les travaux de [126] proposent une solution générale, valable quelle que soit la trajectoire considérée au problème de propagation des erreurs systématiques et stochastiques. Martinelli dans [159] généralise, dans le cas d’un système synchro-drive, la formulation de la matrice de covariance pour une trajectoire quelconque et définit 4 paramètres permettant de caractériser 2 à 2 les erreurs systématiques et non systématiques.

- On peut aussi faire appel, durant le fonctionnement du robot, aux informations redondantes, supposées plus précises, issues d’autres capteurs, pour calibrer les erreurs en faisant appel à des filtres de Kalman [201] [137].

Commençons donc par construire le modèle causes-effets de l’erreur odométrique.

### 5.2.2 Phase 1 : Diagramme Causes-effets

Pour construire le diagramme causes-effets nous allons adapter l’approche développée par Ishikawa dans les années 60 dont l’objectif est de proposer un cadre visuel de réflexion permettant une analyse efficace des dysfonctionnements de processus industriels. Ce dernier a proposé un diagramme en arêtes de poisson où la flèche principale horizontale identifie l’effet indésirable observé, et où des flèches obliques guident l’identification des causes pouvant être à l’origine de cet effet. Classiquement, 5 points de vue complémentaires, les 5M, permettent de balayer l’environnement industriel. Un dysfonctionnement peut trouver ses sources dans la **Matière** transformée, le **Matériel** utilisé, la **Main d’œuvre** employée, la **Méthode** déployée et le **Milieu** supportant le processus industriel étudié. Des flèches secondaires rattachées à ces grandes familles de causes peuvent ensuite venir enrichir la réflexion.

Dans le cadre qui nous intéresse, c’est à dire l’analyse de la performance d’une méthode de Localisation déployée sur une plateforme robotique autonome, la flèche principale correspond à l’indicateur de performance et donc, dans notre cas, à la précision de Localisation. Il est clair que cet indicateur est tri-dimensionnel mais, par esprit de simplification, nous ne les distinguerons pas. Tout d’abord, puisque nous considérons des missions robotiques autonomes, le facteur **Humain** (Main d’œuvre) disparaît naturellement puisqu’il n’influence pas directement la précision de localisation. Pour les autres causes nous adopterons la logique d’analyse suivante. La précision de localisation est impactée par le choix d’une **Méthode** de localisation, faisant appel à un ensemble de **Paramètres** d’entrée (Matière), utilisant des ressources **Matérielles** embarquées ou non, se déployant au sein d’un **Milieu** (Environnement) connu.

En projetant ce cadre de réflexion pour l’analyse de l’erreur de localisation odométrique d’un robot unicycle, on obtient le diagramme d’Ishikawa de la Figure 5.2 au sein duquel les causes qui constituent des inducteurs de performance contrôlables hors ligne sont indiqués en bleu.

La Méthode de calcul de l’erreur odométrique suppose que le système étudié a un comportement linéaire sur une période d’échantillonnage et que les données (paramètres) se comportent aléatoirement comme des distributions gaussiennes dont on a estimé expérimentalement les caractéristiques. Évidemment cela ne constitue qu’une approximation de la réalité. Par ailleurs l’incertitude de localisation dépend de la distance parcourue, qui est elle-même reliée au chemin choisi et à la connaissance plus ou moins précise de la position initiale du robot. Elle est aussi influencée par la qualité des odomètres utilisés, de la fréquence de rafraîchissement des informations odométriques et des erreurs d’actionnement du contrôle de la plateforme. Par ailleurs les incertitudes de connaissance sur le dimensionnement exact des roues et de l’entraxe induira une erreur systématique qu’il faudra chercher à compenser. Enfin le milieu est une source très

importante d'incertitudes en modifiant la qualité du contact roue-sol. De plus la dynamique de l'environnement conduira la trajectoire du robot à s'éloigner de celle prévue ce qui peut impacter la validité du modèle d'incertitude utilisé établi pour une trajectoire donnée.

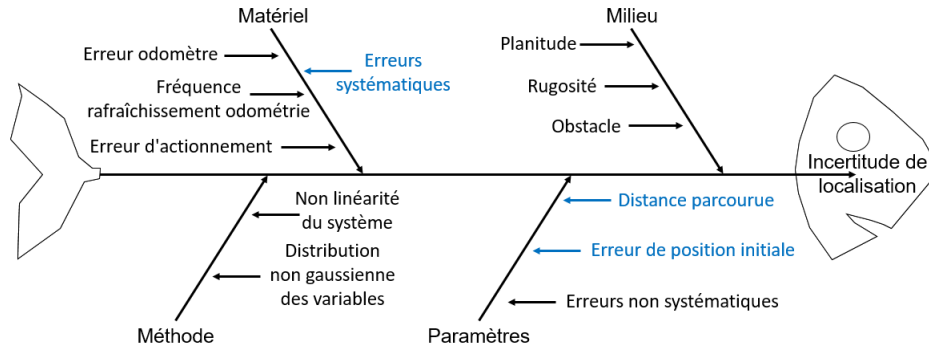
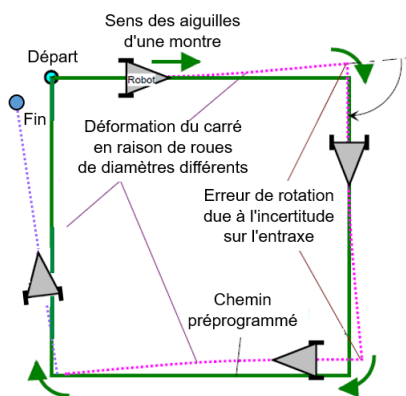


FIGURE 5.2 – Diagramme causes-effets : Erreur de localisation odométrique

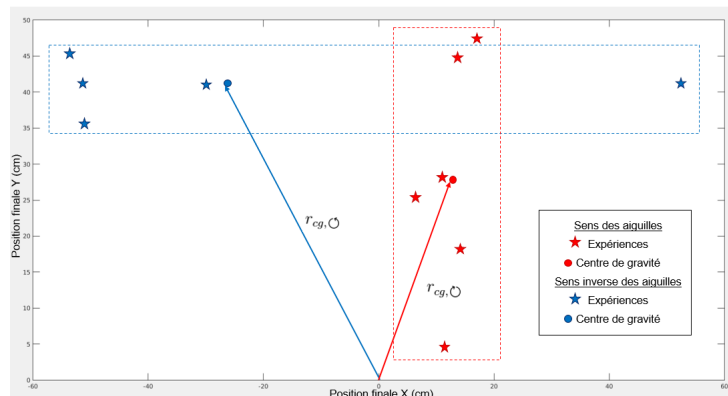
L'erreur odométrique présentant un biais dû aux erreurs systématiques induites par les incertitudes des paramètres cinématiques, nous allons maintenant déployer l'approche UMBmark pour le corriger.

### 5.2.3 Phase 2 : Erreurs systématiques

La méthode développée par Borenstein propose de parcourir une trajectoire fermée (un carré) (Figure 5.3 a)) dans le sens des aiguilles d'une montre  $\curvearrowright$  et dans le sens inverse  $\curvearrowleft$ , puis de reporter sur un graphique l'erreur de localisation obtenue en comparant la position finale observée et celle estimée par l'odométrie du robot (Figure 5.3 b)). L'expérience est répétée plusieurs fois. On constate que les erreurs calculées sont concentrées dans 2 régions distinctes, en fonction du sens de parcours de la trajectoire. En effectuant l'expérience sur une surface plane et lisse on peut admettre que la distribution des erreurs au sein de chacune des régions dépend principalement des erreurs systématiques et son centre de gravité  $cg$  sera considéré comme représentatif des erreurs observées pour chaque sens de parcours.



a) Méthode UMBmark : Principe d'expérimentation



b) Méthode UMBmark : Incertitudes expérimentales systématiques avant correction

FIGURE 5.3 – Méthode UMBmark

Il est alors aisé de définir les coordonnées  $x_{cg}$  de chacun des centres de gravité et l'offset  $r_{cg}$  correspondant par les Équations 5.1 où  $\epsilon X/Y, \curvearrowright, \curvearrowleft$  représentent les erreurs en X et Y mesurées après une rotation.

$$\begin{aligned}
X_{cg,\cup,\mathcal{U}} &= \frac{1}{n} \sum_{i=1}^n \epsilon X_{i,\cup,\mathcal{U}} \\
Y_{cg,\cup,\mathcal{U}} &= \frac{1}{n} \sum_{i=1}^n \epsilon Y_{i,\cup,\mathcal{U}} \\
r_{cg,\cup} &= \sqrt{X_{cg,\cup}^2 + Y_{cg,\cup}^2} \\
r_{cg,\mathcal{U}} &= \sqrt{X_{cg,\mathcal{U}}^2 + Y_{cg,\mathcal{U}}^2}
\end{aligned} \tag{5.1}$$

Avec notre plateforme nous avons parcouru 6 fois un carré de 4 m de côté. Les relevés effectués sont présentés dans le graphique de la Figure 5.3 b). L'application des équations précédentes a conduit aux valeurs expérimentales suivantes (Équations 5.2).

$$X_{cg,\cup} = 128.33 \text{ mm} \quad Y_{cg,\cup} = 278.83 \text{ mm} \quad r_{cg,\cup} = 306.95 \text{ mm}$$

$$X_{cg,\mathcal{U}} = -262.40 \text{ mm} \quad Y_{cg,\mathcal{U}} = 412.80 \text{ mm} \quad r_{cg,\mathcal{U}} = 489.14 \text{ mm} \tag{5.2}$$

Pour compenser ces erreurs systématiques les rapports suivants sont définis (Equation 5.3).

$$E_d = \frac{D_{d_r}}{D_{g_r}} \quad E_b = \frac{b_r}{b_{nom}} \tag{5.3}$$

où  $D_{d_r}$  et  $D_{g_r}$  représentent les diamètres réels des roues droite et gauche respectivement,  $b_r$  la valeur de l'entraxe réel qui peut être différente de la valeur nominale  $b_{nom}$ .  $E_d$  qui caractérise les erreurs induites par des diamètres de roues non nominaux (pneu dégonflé) est la source principale du biais lors de translations.  $E_b$  est le facteur principal d'erreur dans le cas de rotation pure.

En analysant géométriquement le comportement du robot il est alors possible d'établir les équations suivantes [35].

L'angle  $\alpha$  représente la moyenne de l'erreur de rotation effectuée lors de chaque rotation de  $90^\circ$  exécutée à chaque angle du carré après avoir parcouru un côté de longueur  $L$ .

$$\alpha = \frac{(X_{cg,\cup} + X_{cg,\mathcal{U}})}{-4L} \cdot \frac{180}{\pi} = \frac{(Y_{cg,\mathcal{U}} - Y_{cg,\cup})}{-4L} \cdot \frac{180}{\pi} \tag{5.4}$$

Une fois  $\alpha$  déterminé à partir des expérimentations on peut calculer l'entraxe réel  $b_r$ .

$$b_r = \frac{90^\circ}{90^\circ - \alpha} \cdot b_{nom} \tag{5.5}$$

On peut aussi calculer  $\beta$ , l'erreur d'orientation moyenne à la fin de parcours d'un côté du carré de longueur  $L$ .

$$\beta = \frac{(Y_{cg,\cup} + Y_{cg,\mathcal{U}})}{-4L} \cdot \frac{180}{\pi} = \frac{(X_{cg,\mathcal{U}} - X_{cg,\cup})}{-4L} \cdot \frac{180}{\pi} \tag{5.6}$$

Une fois  $\beta$  déterminé à partir des expérimentations, il est possible de calculer le rayon de courbure  $R$  du cercle suivi expérimentalement lors d'une translation désirée en ligne droite.

$$R = \frac{\frac{L}{2}}{\sin(\frac{\beta}{2})} \tag{5.7}$$



On peut alors aisément estimer la valeur du rapport  $E_d$ .

$$E_d = \frac{D_{dr}}{D_{gr}} = \frac{R + \frac{b_r}{2}}{R - \frac{b_r}{2}} \quad (5.8)$$

Ce qui permet de déduire les valeurs de  $D_{dr}$  et de  $D_{gr}$

$$D_{gr} = \frac{2}{E_d + 1} \cdot D_a = C_g \cdot D_a \quad D_{dr} = \frac{2}{\frac{1}{E_d} + 1} \cdot D_a = C_d \cdot D_a \quad (5.9)$$

Où  $D_a = \frac{D_{gnom} + D_{dnom}}{2}$  représente le diamètre moyen des roues qui est supposé inchangé et où  $C_d$  et  $C_g$  représentent les facteurs de correction à appliquer sur les roues droite et gauche respectivement pour compenser les erreurs systématiques.

Avec nos relevés nous avons obtenu les valeurs expérimentales suivantes :

$$\begin{aligned} b_r &= 35.64 \text{ cm} & D_{gr} &= 19.00 \text{ cm} & D_{dr} &= 19.00 \text{ cm} \\ C_g &= 0.99996289 & C_d &= 1.00003710 \end{aligned} \quad (5.10)$$

Une fois les erreurs systématiques compensées nous pouvons maintenant nous focaliser sur celles qui ne le sont pas.

### 5.2.4 Phase 3-1 : Modèle d'erreur odométrique de position

Pour établir le modèle d'erreur odométrique nous allons nous appuyer sur le raisonnement détaillé dans le chapitre 5 de [204] pour un robot unicycle que nous allons résumer ici puis décliner sur notre plateforme.

L'évolution de la pose d'un robot unicycle peut s'exprimer par l'équation récurrente 5.11 où  $\Delta x, \Delta y, \Delta \theta$  représentent les incréments de pose, et  $\Delta S_g, \Delta S_d$  les incréments de déplacement des roues gauche et droite, entre 2 instants d'échantillonnage amenant le robot d'une position  $p$  à une position  $p'$ .

$$p' = \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = p + \begin{pmatrix} \Delta S \cdot \cos(\theta + \frac{\Delta \theta}{2}) \\ \Delta S \cdot \sin(\theta + \frac{\Delta \theta}{2}) \\ \Delta \theta \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \Delta S \cdot \cos(\theta + \frac{\Delta \theta}{2}) \\ \Delta S \cdot \sin(\theta + \frac{\Delta \theta}{2}) \\ \Delta \theta \end{pmatrix} \quad (5.11)$$

$$\text{Où } \Delta S = \frac{\Delta S_d + \Delta S_g}{2} \text{ et } \Delta \theta = \frac{\Delta S_d - \Delta S_g}{b}$$

De par la récurrence et les incertitudes de déplacement  $\Delta S_d$  et  $\Delta S_g$ , l'incertitude sur la position de  $p$  va s'accroître dans le déplacement du robot. Une approche statistique va donc être adoptée de façon à pouvoir estimer la matrice de covariance  $\sum_{p'}$  de  $p'$  en sortie d'un système  $p' = f(x, y, \theta, \Delta S_g, \Delta S_d)$  dont la fonction  $f$  est connue ainsi que la matrice initiale de covariance  $\sum_p$ . On supposera évidemment que toutes les variables, tant en entrée qu'en sortie, sont des variables aléatoires de distribution gaussienne.

En supposant que  $p$  et  $\Delta_{dg} = (\Delta S_d; \Delta S_g)$  sont non corrélés il est alors possible d'exprimer la covariance de  $p'$  par l'Équation 5.12 en appliquant le théorème de propagation des erreurs.

Ce dernier énonce que pour un système comportant  $n$  entrées  $X_i$  de distribution de probabilité connue et  $m$  sorties  $Y_i$ , si on approxime la fonction  $f$  connue par un développement de Taylor au 1<sup>er</sup> ordre (linéarisation) alors  $C_Y = F_X \cdot C_X \cdot F_X^T$  où  $F_X = \nabla f$  est une matrice jacobienne,  $C_X$  la matrice de covariance des entrées et  $C_Y$  celle associée aux sorties.

$$\sum_{p'} = \nabla_p f \cdot \sum_p \cdot \nabla_p f^T + \nabla_{\Delta dg} \cdot \sum_{\Delta} \cdot \nabla_{\Delta dg}^T \quad (5.12)$$

Dès lors, si l'on suppose que l'on connaît la matrice de covariance initiale  $\sum_p$  il est possible de calculer par récurrence la matrice de covariance de sortie  $\sum_{p'}$  entre 2 instants d'échantillonnage à condition de connaître la matrice de covariance  $\sum_{\Delta}$  liée aux incertitudes de déplacement. Pour y parvenir les hypothèses suivantes sont adoptées en accord avec [204] :

- Les erreurs de déplacement sont indépendantes l'une de l'autre. Cela se justifie par le fait que les 2 roues sont commandées par des moteurs différents et utilisent des encodeurs séparés [61] [35].
- Pour de petits déplacements, les erreurs sont supposées nulles et de bruit blanc. La variance de l'erreur cumulée peut donc s'exprimer comme étant la somme des variances indépendantes des déplacements. On peut donc postuler que, pour chaque roue, la variance des erreurs sera proportionnelle à la valeur absolue du déplacement de la roue considérée [61]. Les facteurs de proportionnalité seront donc des constantes  $k_d$  et  $k_g$  dépendantes des moteurs utilisés et du contact roue-sol et s'exprimant en  $m$ .

Ainsi on peut donc décrire  $\sum_{\Delta}$  par l'Équation 5.13.

$$\sum_{\Delta} = \begin{pmatrix} k_d \cdot |\Delta S_d| & 0 \\ 0 & k_g \cdot |\Delta S_g| \end{pmatrix} \quad (5.13)$$

Les constantes de proportionnalité  $k_d$  et  $k_g$  dépendant de la plateforme robotique et de l'environnement, il faut donc les déterminer expérimentalement. Nous allons maintenant présenter comment nous les avons calculées.

Nous avons utilisé la zone d'expérimentation du couloir 2 équipée d'un laser externe fixe dans l'environnement pour repérer la position de notre plateforme robotique qui décrira une trajectoire rectiligne en boucle ouverte, à vitesse constante, sur 15 m, dans sa direction. Nous pourrons donc avoir 2 estimations de la position du robot. L'une sera déduite des mesures réalisées par le laser externe, échantillonnant à la fréquence de 40 Hz, que nous considérerons comme notre référence terrain, et que nous supposerons sans erreur. L'autre sera issue de la lecture de l'odométrie effectuée directement sur la plateforme et échantillonnée à la fréquence de 10 Hz. Ces mesures sont datées et synchronisées temporellement.

Des mesures de position réalisées on déduit, entre 2 instants d'échantillonnage synchrones, la vitesse de chacune des roues du robot. La différence de vitesse entre la mesure laser et robot nous permet de calculer l'erreur d'estimation de vitesse correspondante. La moyenne sur nos échantillons, normalisée par la vitesse de consigne choisie, nous permet d'obtenir les coefficients  $k_d$  et  $k_g$  recherchés grâce à l'Équation 5.14. Une trajectoire rectiligne a été choisie puisque c'est celle qui est déployée au cours de notre expérimentation cible.

$$k_{d/g} = \frac{\text{Variance}((\Delta S_{las})_{d/g} - (\Delta S_{odo})_{d/g})}{\text{Moyenne}(\Delta S_{Consigne})_{d/g}} \quad (5.14)$$

Nous avons ainsi trouvé les valeurs expérimentales suivantes :

$$k_d = 0.0004523 \text{ m} \quad k_g = 0.0004522 \text{ m} \quad (5.15)$$

Il est alors possible en utilisant l'équation récurrente 5.12 de déterminer l'ellipse d'incertitude à la matrice de covariance  $\Sigma_p$ , après chaque échantillonnage. La longueur des 2 demi-axes correspond à la racine carrée des valeurs propres les plus grandes  $\lambda_1$  et  $\lambda_2$  qui représentent en fait des variances. Avec des variables gaussiennes, pour obtenir une probabilité de position garantie à 99.7 %, la longueur d'un demi-axe vaudra  $3 \cdot \sqrt{\lambda_i}$ .

### 5.2.5 Phase 3-3 : Incertitude de position : Évaluation

Pour une trajectoire perçue donnée il est possible de tracer l'évolution de l'ellipse d'incertitude correspondant au modèle d'erreur odométrique en position. La Figure 5.4 la représente pour une trajectoire rectiligne. On constate classiquement que l'incertitude dans le sens du déplacement croît beaucoup moins vite que celle estimée perpendiculairement. Grâce à l'Équation 5.12 et à la connaissance de la distance à parcourir il est alors possible de déterminer la dimension du pavé d'incertitude garantie en position. Le terme garantie est cependant un léger abus de langage et nous supposerons comme dans [140] qu'une probabilité à  $3\sigma$  est suffisante pour pouvoir l'adopter.

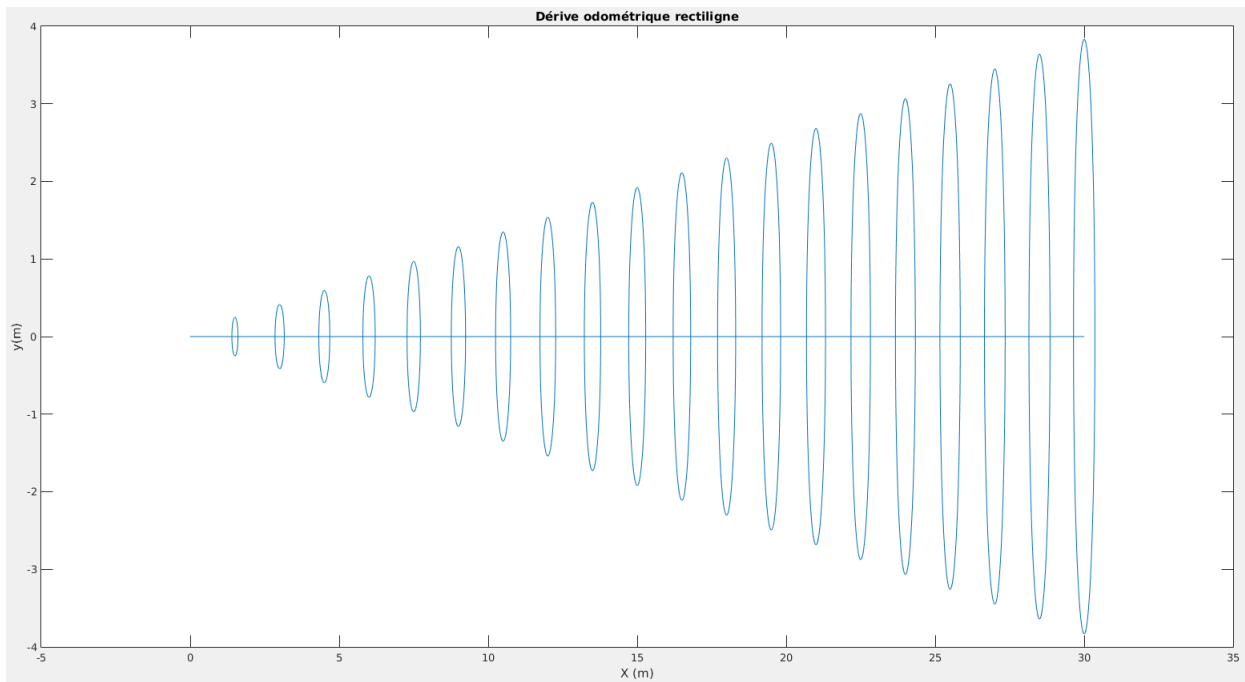


FIGURE 5.4 – Évolution de l'erreur odométrique pour une trajectoire rectiligne

Curieusement, à notre connaissance, on trouve peu de travaux se rapportant à l'évolution de l'incertitude en rotation et de représentation de celle-ci. Nous allons donc maintenant détailler la démarche que nous avons suivie.

### 5.2.6 Phase 3-1 : Modèle d'erreur odométrique de rotation

Pour estimer l'incertitude en orientation du robot nous allons nous appuyer sur l'Équation 5.16 la définissant à partir de l'expression de la distance parcourue par chacune des roues et de la connaissance de l'entraxe.

$$\Delta\theta = \frac{\Delta S_d - \Delta S_g}{b} \quad (5.16)$$

Pour déterminer l'incertitude de localisation nous avons déjà identifié les valeurs des coefficients de proportionnalité  $k_d$  et  $k_g$  de chacune des roues qui correspondent aux variances des erreurs de déplacement observées. En nous inspirant du raisonnement proposé dans [61] [159] on peut donc considérer que si l'on suppose un comportement gaussien, il est possible de représenter la distribution des valeurs de  $k_{d/g}$  par une variable aléatoire  $SK_{d/g}$  suivant une loi normale  $\mathcal{N}(\mu, \sigma^2)$  de l'Équation 5.17 de valeur moyenne  $\mu$  et variance  $\sigma^2$ .

$$SK_{d/g} \sim \mathcal{N}(\Delta S_{d/g}, k_{d/g} \Delta S_{d/g}) \quad (5.17)$$

En introduisant ces lois normales dans l'expression de  $\Delta\theta$  de l'Équation 5.16 on peut traduire le comportement aléatoire gaussien des déplacements de chacune des roues autour de sa position moyenne  $\Delta S_{d/g}$  par l'équation 5.18.

$$\Delta\theta \sim \frac{SK_d - SK_g}{b} \quad (5.18)$$

Nous sommes alors face à une composition linéaire de 2 lois normales  $\mathcal{N}_1(\mu_1, \sigma_1^2)$  et  $\mathcal{N}_2(\mu_2, \sigma_2^2)$  de variables supposées indépendantes qui conserve par additivité le caractère gaussien pour obtenir une loi  $\mathcal{N}_{12}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$ . Il est alors aisé de conclure que dans notre cas on obtient l'Équation 5.19.

$$\Delta\theta \sim \mathcal{N}\left(\frac{\Delta S_d - \Delta S_g}{b}, \frac{k_d \cdot \Delta S_d + k_g \cdot \Delta S_g}{b^2}\right) \quad (5.19)$$

De la même façon que pour l'incertitude de position il est maintenant possible de déterminer par simulation l'évolution de l'incertitude en  $\theta$  pour une trajectoire donnée. Dans le cas de la trajectoire rectiligne qui nous intéresse on obtient la courbe de la Figure 5.5.

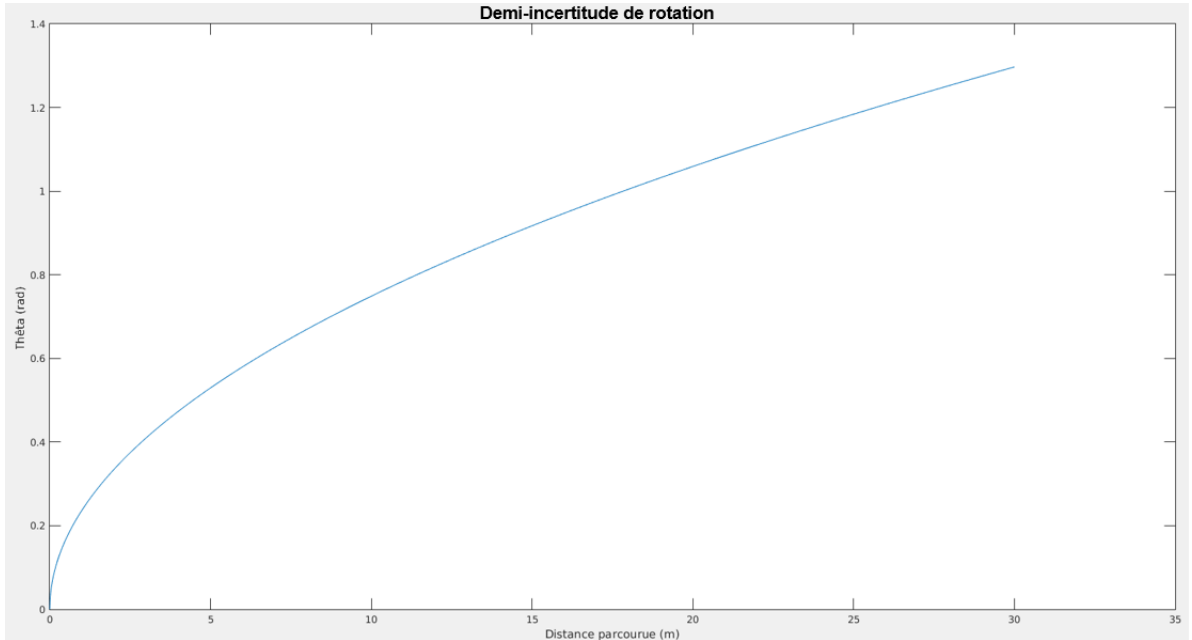


FIGURE 5.5 – Évolution de l'incertitude d'orientation odométrique pour une trajectoire rectiligne

On peut observer que ce modèle prévoit un maintien du cap à  $\pm \pi/4$  sur 11 m et même à  $\pm \pi/8$  pendant 3.75 m. Cela est suffisant pour garantir une connaissance acceptable de l'orientation si l'on utilise l'odométrie sur des distances raisonnables.

## 5.2.7 Phases 3-3 et 4 : Incertitude de pose : Évaluation et Validation

La Figure 5.6 représente d'une part l'incertitude de pose maximale déduite des modèles précédents ainsi que les erreurs de mesures relevées expérimentalement. On constate que ces dernières restent très largement comprises à l'intérieur des limites théoriques obtenues. On peut aussi remarquer que nous ne pouvons pas calculer nos erreurs de mesures lorsque les incertitudes de notre réalité terrain deviennent supérieures à celles de nos modèles théoriques.

Il faut tout de même souligner que la mesure de la pose du robot a été obtenue en prenant la valeur médiane de l'intersection entre le pavé d'incertitude correspondant à notre laser externe, et celui de notre Kinect 2. Dès lors, notre incertitude de mesure de réalité terrain est très largement supérieure aux erreurs de mesure de déplacement de l'odométrie. Cela souligne que cette dernière, une fois les erreurs statiques compensées, est de bonne qualité puisque au bout de 15 m, la dérive en X est d'environ  $\pm 0.1$  m et celle en Y de l'ordre de 0.5 m, à priori centrée sur une erreur statique trop faible pour pouvoir être corrigée par la commande du Pioneer. Une meilleure réalité terrain aurait permis d'obtenir un modèle théorique d'incertitude moins conservatif.

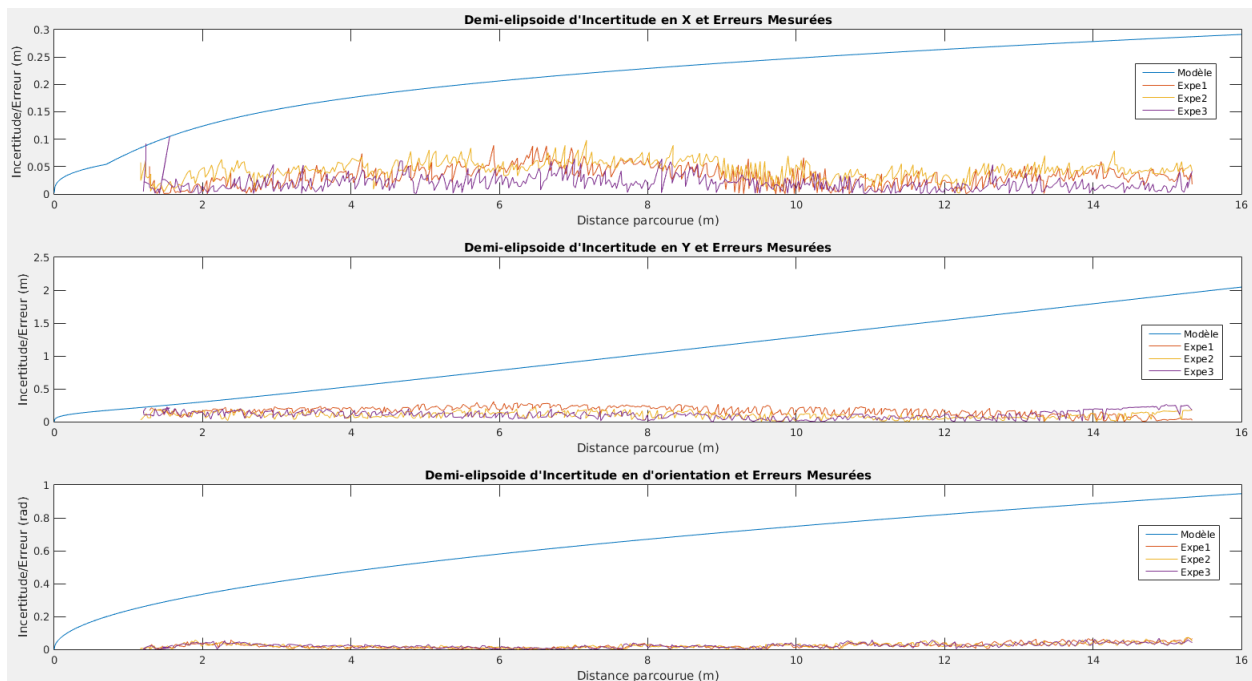


FIGURE 5.6 – Comparaison entre l'erreur odométrique observée et celle du modèle pour une trajectoire rectiligne

Ces résultats expérimentaux confirment donc que nous pouvons utiliser les modèles théoriques odométriques établis pour définir le modèle intervalliste prédictif de pose garantie que nous recherchons.

## 5.2.8 Conclusion

Cette partie a permis d'établir un modèle de dérive odométrique pour notre plateforme robotique. Nous avons suivi une démarche classique s'appuyant sur l'identification des erreurs systématiques faisant appel à la démarche UMBmark proposée par Borenstein, et sur l'analyse de la matrice de covariance pour identifier les paramètres du modèle d'erreur aléatoire associé aux erreurs non-systématiques. Les calibrations effectuées nous ont permis d'obtenir un modèle d'erreur de la garantie odométrique pouvant être adoptée pour une trajectoire rectiligne. Ce

travail se traduit pratiquement par un bien meilleur comportement de notre odométrie que celui remarqué dans [112] où une dérive latérale de près de 50 cm était observée pour une trajectoire rectiligne d'une douzaine de mètres. Il n'en demeure pas moins que l'ellipse d'incertitude de localisation croît avec la distance parcourue et cela encore plus rapidement perpendiculairement au sens de déplacement du robot ce qui induit aussi une incertitude de cap croissante mais restant acceptable sur des distances raisonnables. Rien n'assure que l'incertitude observée explosera toujours rapidement expérimentalement avec la distance parcourue (statistique), mais malheureusement rien ne certifie qu'elle ne le fera pas non plus (glissements, trajectoire non rectiligne, etc.).

### 5.3 KIN-ARUCO : Localisation extéroceptive par amers artificiels verticaux

Dans cette partie nous allons établir le modèle d'incertitude de localisation lié à la lecture d'amers géoréférencés Aruco par la caméra Kinect 1 de notre plateforme. Pour ce faire nous allons tout d'abord présenter le diagramme d'Ishikawa correspondant avant de préciser comment le modèle théorique peut être construit à partir des résultats obtenus dans le chapitre précédent. Pour finir, nous allons le comparer avec les résultats expérimentaux relevés dans notre zone d'expérimentation du couloir 2.

#### 5.3.1 Phase 1 : Diagramme Causes-Effets

Le diagramme d'Ishikawa de la Figure 5.7 permet d'identifier les éléments impactant l'incertitude de la méthode KIN-ARUCO.

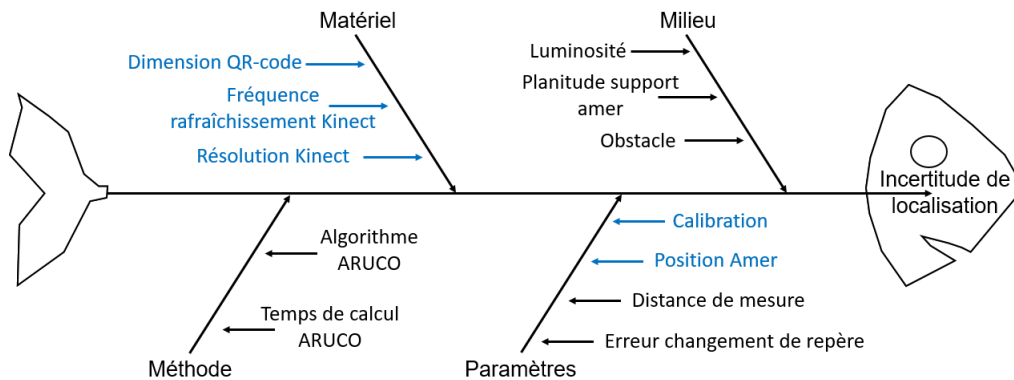


FIGURE 5.7 – Diagramme causes-effets : Erreur de localisation Kin-Aruco

Cette incertitude va dépendre d'un ensemble de paramètres. Tout d'abord la position des amers dans l'environnement. Évidemment en multipliant les amers lorsque cela est possible on réduira l'incertitude de pose. Par ailleurs elle dépend aussi de la qualité de la calibration effectuée. Dans notre cas cela est lié à la fois à la Kinect elle-même, mais aussi à la méthode Aruco qui permet d'exécuter cette calibration pour une cible de dimension donnée. Évidemment la distance de la Kinect à l'amer, qui évoluera dynamiquement en fonction du déroulement de la mission et de la position des amers dans l'environnement, aura une incidence très importante sur l'erreur de pose induite. Pour terminer l'erreur de changement de repère intra-plateforme permettant de passer du repère Kinect au repère Robot aura aussi un impact constant mais limité.

En ce qui concerne le matériel utilisé, il est évident que la dimension de l'amer influence l'incertitude. Il y a un compromis à avoir entre la possibilité de voir la cible de loin, et la pré-

cision de localisation que l'on peut espérer obtenir en fonction de la pixelisation de l'image. La résolution de la caméra, ainsi que sa fréquence de rafraichissement, auront un impact important. Mais la fréquence de rafraichissement va aussi être impactée par la vitesse du robot qui, en raison des irrégularités du sol, va faire légèrement osciller le support de la Kinect, réduisant la fréquence d'obtention d'images exploitables. C'est pour cela que nous limitons notre vitesse de déplacement lorsque la Kinect est employée à 0.5 m/s.

La luminosité du milieu va fortement influencer la qualité des images. La planéité parfois relative des supports des amers peut induire des estimations de rotations inopportunes perturbant significativement la lecture de l'orientation et donc plus globalement l'estimation de la pose du robot. Enfin, de la même façon, le changement de trajectoire du robot en présence d'obstacles peut compliquer l'observation des amers par la Kinect.

Pour terminer, la mise en œuvre matérielle de l'algorithme Aruco va induire un temps de calcul impactant la durée séparant l'obtention d'une image Kinect et le calcul de la pose du robot.

En considérant que les calibrations de la Kinect et d'Aruco procèdent de la phase 2 et de l'élimination d'erreurs systématiques, on peut directement passer à la définition du modèle théorique Kinect-Aruco.

### 5.3.2 Phase 3-1 : Modèle d'erreur théorique

Pour établir le modèle théorique de la méthode Kinect-Aruco nous allons nous appuyer sur la formulation générique de l'expression de la pose du robot dans le repère Monde à partir de la lecture d'un amer dans le repère Kinect et de la matrice de transformation de repère Kinect vers le repère Robot.

L'Equation 4.3 exprime le lien entre  $\mathcal{R}_R(t)$  à  $\mathcal{R}_{Aruco}$ , où  $MT_{\mathcal{R}_{kinect}(t)|\mathcal{R}_{Aruco}}$  exprime la matrice de transformation du repère Aruco au repère Kinect, et  $MT_{\mathcal{R}_R|\mathcal{R}_{kinect}}$  la matrice de changement de repère intra-plateforme. Les pavés associés à ces 2 matrices ont été caractérisés dans le chapitre précédent.

Donc si l'on veut relier  $\mathcal{R}_R(t)$  à  $\mathcal{R}_{Aruco}$  on peut écrire l'Équation 5.20 suivante :

$$\begin{bmatrix} O_{\mathcal{R}_{Aruco}|\mathcal{R}_R(t)} \end{bmatrix} = MT_{\mathcal{R}_R|\mathcal{R}_{kinect}} \bullet MT_{\mathcal{R}_{kinect}(t)|\mathcal{R}_{Aruco}} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (5.20)$$

Maintenant pour exprimer la position du robot dans le repère Monde on peut utiliser l'Équation 5.21, où la matrice  $MT_{\mathcal{R}_R(t)|\mathcal{R}_{Aruco}}^{-1}$  correspond à la transformation inverse de l'équation précédente.

$$\begin{bmatrix} O_{\mathcal{R}_R(t)|\mathcal{R}_M} \end{bmatrix} = MT_{\mathcal{R}_M|\mathcal{R}_{Aruco}} \bullet MT_{\mathcal{R}_R(t)|\mathcal{R}_{Aruco}}^{-1} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (5.21)$$

Il faut rajouter évidemment l'Equation 4.6 permettant de déduire l'erreur l'orientation du robot.

$$\theta_{\mathcal{R}_M|\mathcal{R}_R} = \theta_{\mathcal{R}_M|\mathcal{R}_{Aruco}} + \theta_{\mathcal{R}_{Aruco}|\mathcal{R}_{kinect}} + \theta_{\mathcal{R}_{kinect}|\mathcal{R}_R} \quad (5.22)$$

A partir de ces trois équations et de la connaissance des pavés d'incertitude sur la position du marqueur Aruco, de la lecture Kinect et de la transformation de référentiel, il est possible d'obtenir le pavé résultant en utilisant les relations ensemblistes de l'algèbre par intervalle.

### 5.3.3 Phases 3-3 et 4 : Relevés expérimentaux et évaluation

Pour s'assurer que le modèle théorique défini plus haut est valide nous avons exécuté avec notre plateforme, dans notre zone d'expérimentation habituelle du couloir 2, plusieurs trajectoires rectilignes en boucle ouverte pour des vitesses allant de 0.3 m/s à 0.5 m/s. Les marqueurs Aruco ont été disposés régulièrement de part et d'autre de la trajectoire à 1.2 m. Nous avons utilisé le laser 30 m à notre disposition, ainsi que la Kinect 2 pour obtenir la réalité terrain tout au long de la zone d'expérimentation. Grâce aux mesures synchrones effectuées au cours d'une expérimentation nous avons pu établir un modèle expérimental de pose du QR-Code dans le repère Robot, pour des marqueurs situés à une distance comprise entre 1 et 5 m, au-delà les incertitudes deviennent vraiment trop importantes. Ce relevé, en rouge dans la Figure 5.8, doit s'inscrire à l'intérieur des limites du modèle théorique de l'Équation 4.3 tracées en bleu. On peut remarquer que les courbes d'incertitude d'orientation théorique et expérimentale sont quasiment confondues car, pour le calcul théorique de l'incertitude de position, le relevé dynamique n'est impacté que par la légère erreur d'orientation de la caméra lors du changement de repère.

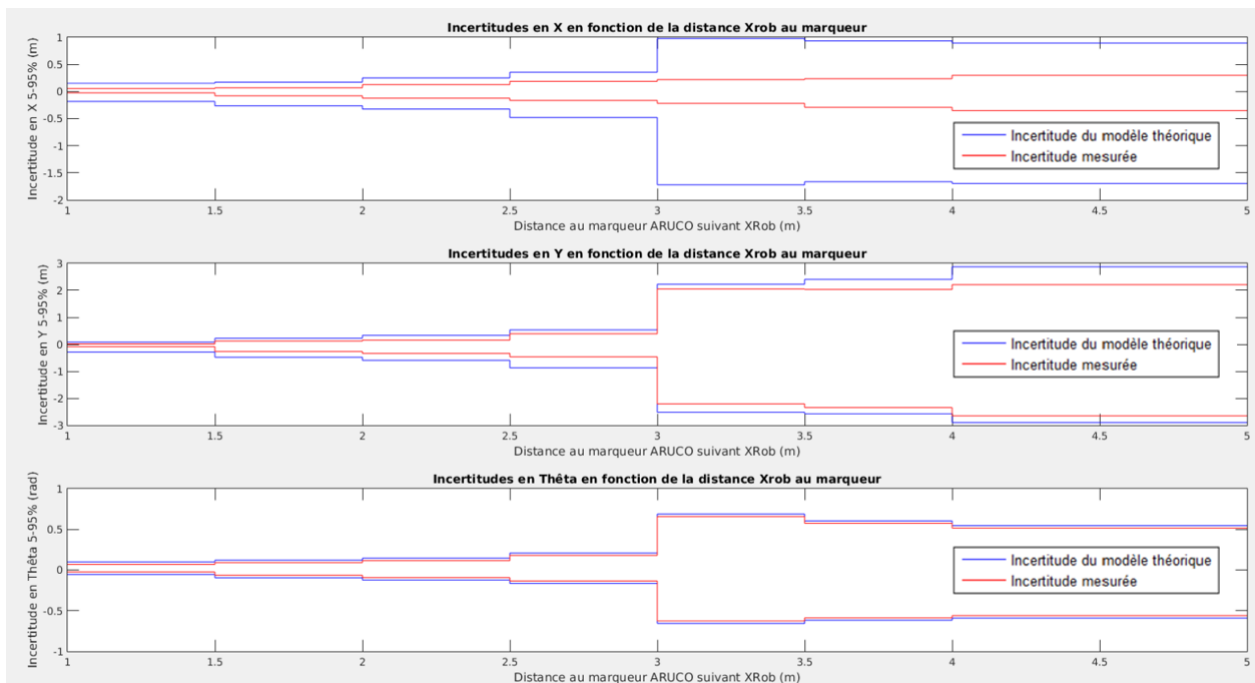


FIGURE 5.8 – Incertitude de pose du modèle Kinect-Aruco : Comparaison des résultats expérimentaux avec le modèle théorique

On peut constater sur ces figures que les courbes expérimentales sont toutes incluses à l'intérieur des courbes théoriques. Cela permet donc de conclure à la validité du modèle théorique établi. On observe que la courbe expérimentale en X reste très éloignée des bornes théoriques mais l'incertitude devient tout de même vraiment importante au delà de 3 m. Malheureusement nous serons obligés dans nos modèles prédictifs d'utiliser le modèle théorique de garantie. L'incertitude observée en Y, quant-à-elle, reste voisine de l'incertitude théorique quelle que soit la distance à l'amer considéré. Pour terminer l'incertitude en  $\theta$  est aussi très importante au delà de 3 m. En raison de son impact sur l'incertitude de position en Y du robot, elle constitue de fait un des facteurs majeurs à l'origine de la détérioration de la performance selon Y. Il nous faudra donc trouver un moyen permettant de limiter notre incertitude angulaire. En conséquence compte tenu de la distance minimale de détection adoptée (2m), et de la distance maximale d'utilisation qui vient d'être énoncée (3m) nous retiendrons pour nos calculs de performance de



localisation que la méthode KIN-ARUCO n'est utilisable que lorsque le robot se trouve à une distance comprise en 2 et 3m du QR-Code.

### 5.3.4 Conclusion

Nous venons d'établir le modèle d'incertitude Kinect-Aruco que nous utiliserons pour nous recalibrer périodiquement, en absolu, lors de notre mission de patrolling en utilisant les amers que nous avons positionnés dans notre environnement. Le modèle théorique a été obtenu en combinant le modèle d'incertitude dynamique établi dans le repère Caméra avec l'incertitude de changement de référentiel dû au passage du repère Caméra au repère Robot. Ce modèle théorique, établi pour des distances comprises entre 1 et 5 m pour des plages de 50 cm diverge notablement lorsque le QR-Code est situé au delà de 3 m du robot. Les expérimentations réalisées ont démontré que le modèle théorique englobe bien les relevés expérimentaux. Cependant nous ne pouvons nous contenter d'une incertitude de pose aussi importante induite par une incertitude d'orientation trop grande. Pour y pallier, nous avons donc mis en œuvre un compas logiciel à partir des relevés de mesures laser dont nous pouvons disposer.

## 5.4 COMPAS : une méthode de détermination de l'orientation

Pour déterminer l'orientation du robot nous allons nous appuyer sur une méthode itérative non déterministe RANSAC [81] (RANdom SAmple Concensus) qui permet d'estimer les paramètres de modèles mathématiques à partir de données expérimentales pouvant contenir des informations aberrantes.

Dans un premier temps nous allons présenter comment la méthode RANSAC est utilisée pour déterminer l'angle d'orientation du robot à partir des informations laser. Puis, dans un second temps, l'analyse du diagramme causes-effets sera détaillée avant d'estimer l'incertitude de l'approche proposée.

### 5.4.1 Description de la méthode COMPAS

La méthode mise en œuvre exploite 2 caractéristiques de notre environnement :

- Le robot se déplace dans un environnement structuré, construit par l'Homme, composé de murs droits généralement rectilignes et perpendiculaires entre-eux.
- Les côtés du couloir étant rectilignes ils sont donc normalement parallèles à la direction du couloir parcouru.

Nous allons donc dans un premier temps, chercher à déterminer la direction relative du couloir par rapport au robot. Puis, dans un second temps, en se basant d'une part sur l'estimation de pose du robot, nous localisant et nous permettant de connaître notre sens de parcours, et d'autre part, sur notre connaissance de l'environnement et donc de l'orientation des couloirs, nous allons pouvoir déterminer l'orientation du robot dans le repère Monde.

De fait, expérimentalement, l'environnement n'est pas toujours composé de couloirs rectilignes et perpendiculaires (portes coupe-feu). L'algorithme RANSAC qui pourrait nous permettre d'extraire directement l'équation des droites des 2 murs du couloir parcouru à partir des informations laser ne peut être directement exploité. Nous procédons donc de la façon suivante (Figure 5.9 a)) :

1. Suppression des points lasers situés en dehors des limites de fonctionnement ou correspondant à des codes d'erreur
2. Utilisation plusieurs fois d'un algorithme de type RANSAC pour déterminer les équations des droites présentes dans le relevé à partir de points choisis aléatoirement, ainsi que leur pondération (nombre de points participant au calcul).
3. Regroupement les pondérations des droites précédentes ayant approximativement la même direction sous l'équation de droite initialement la plus représentée du groupe considéré.
4. L'orientation la plus représentée sera considérée comme la direction du couloir

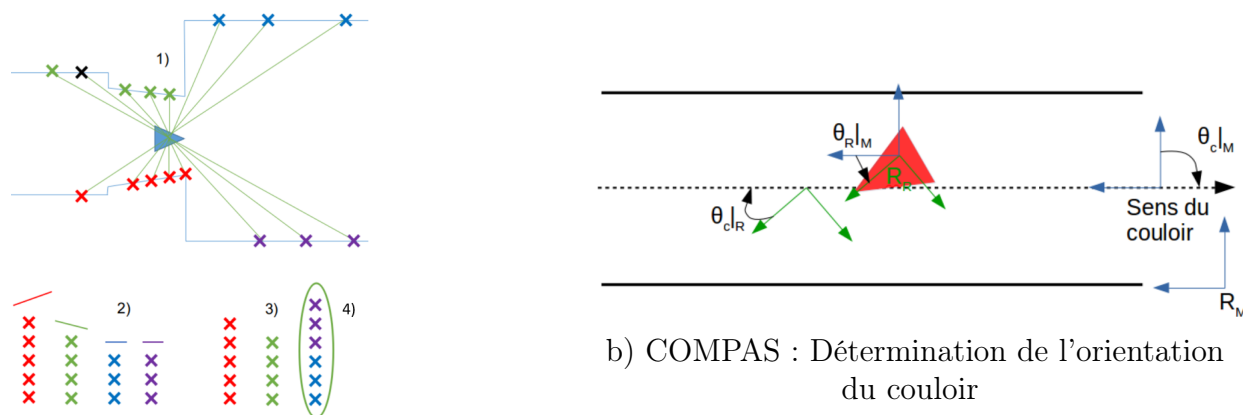


FIGURE 5.9 – COMPAS : Principes méthodologiques

Nous pouvons donc déterminer l'orientation relative du robot par rapport à la direction du couloir. Pour pouvoir lever cette indétermination ( $\{0, \pi\}$ ) nous allons utiliser l'estimation courante de la pose du robot. En s'appuyant sur la Figure 5.9 b), on peut obtenir l'orientation du robot dans le repère Monde en utilisant l'Équation suivante :

$$\theta_{R|M} = (\theta_{c|M} - \theta_{c|R}) + k \cdot \pi \quad \text{avec } k \in \{0, 1\} \quad (5.23)$$

Où  $\theta_{R|M}$  représente l'orientation absolue du robot,  $\theta_{c|M}$  l'orientation du couloir dans le repère Monde qui provient de notre connaissance de l'environnement, et où  $\theta_{c|R}$  correspond à l'orientation du couloir dans le repère Robot déterminée par l'algorithme précédent. Le coefficient  $k$  permet de prendre en compte le sens de déplacement du robot le long du couloir,  $+1$  signifie que le sens de parcours du robot est en sens inverse de l'orientation du couloir dans le repère Monde.

Notre méthodologie de détermination de l'orientation du robot à l'aide des mesures laser étant précisée nous allons maintenant estimer son incertitude.

## 5.4.2 Méthode COMPAS : Modèle d'incertitude de localisation

### 5.4.2.1 Phase 1 : Diagramme Causes-Effets

La Figure 5.10 présente le diagramme d'Ishikawa de la méthode COMPAS.

Bien évidemment l'incertitude d'orientation va dépendre des paramètres choisis pour l'algorithme RANSAC [81] (tolérance distance au modèle évalué et nombre d'évaluations considérées). Elle sera aussi impactée par nos incertitudes de connaissance sur la position courante du robot et sur l'orientation de l'environnement à l'instant considéré.

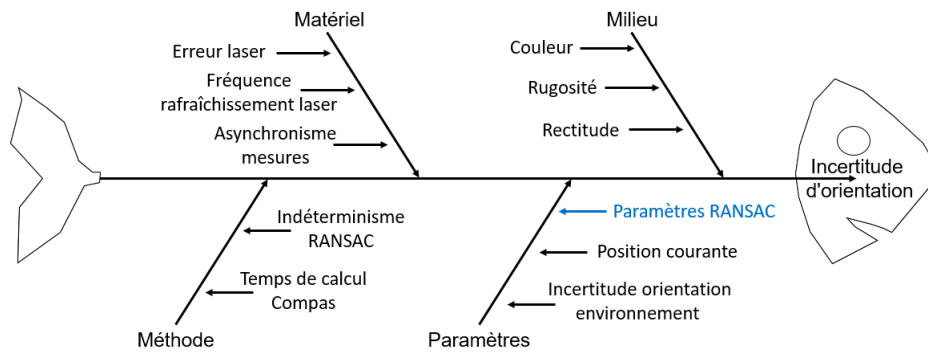


FIGURE 5.10 – Diagramme causes-effets : Erreur d'orientation Compas

Pour un capteur laser donné, l'incertitude d'orientation va dépendre de la précision de ses mesures et de la fréquence à laquelle elles sont obtenues. Comme toujours, le déplacement du robot et l'asynchronisme de mesure de chacun des points du relevé induit une erreur angulaire qui de fait ne sera pas considérée mais qui impactera nécessairement la qualité du relevé.

Évidemment l'environnement va influencer la qualité des relevés lasers (couleur et rugosité des murs). L'algorithme RANSAC qui sera employé pour rechercher des droites devra pouvoir les identifier correctement. Il faut donc que l'environnement ne soit pas trop courbe pour qu'il puisse les extraire.

L'algorithme RANSAC est par essence non déterministe, il existera donc nécessairement une incertitude induite par ce non déterminisme. De plus, comme pour tout algorithme, le déplacement du robot pendant son exécution induira une incertitude sur l'estimation de l'orientation.

#### 5.4.2.2 Phase 3-3 : Construction expérimentale du modèle d'incertitude

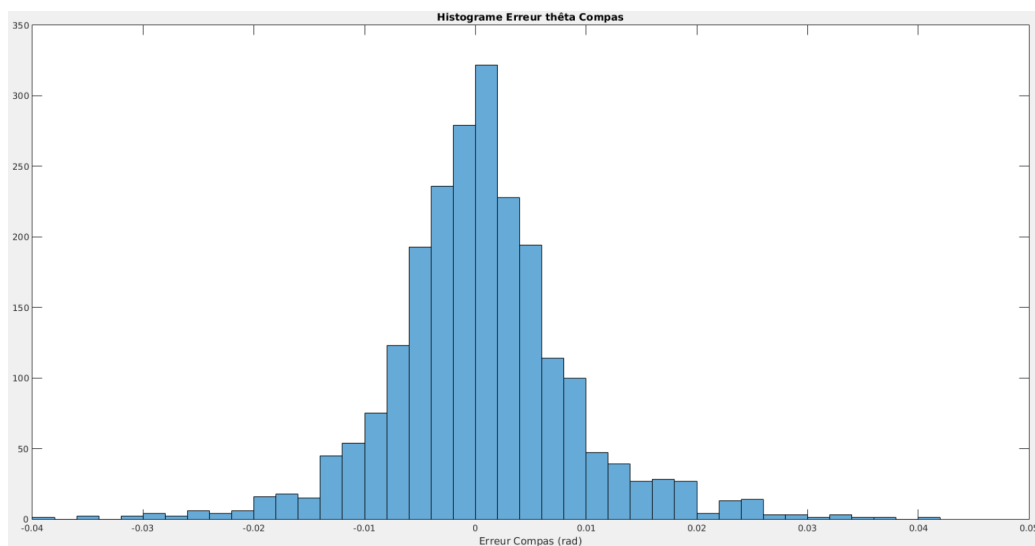


FIGURE 5.11 – COMPAS : Histogramme des incertitudes observées

En l'absence de moyens permettant de caractériser de façon extéroceptive l'erreur d'estimation faite par la méthode COMPAS, l'algorithme a été lancé deux fois sur une série de relevés expérimentaux identiques. Les résultats de la première estimation d'orientation ont servi de référence pour évaluer l'incertitude du second relevé. La Figure 5.11 présente l'histogramme de l'incertitude d'orientation obtenue.

Cet histogramme est caractérisée par une valeur moyenne de  $0.00047 \text{ rad}$ , un écart type  $\sigma = 0.00825 \text{ rad}$ , ce qui correspond à un intervalle d'incertitude de  $\pm 0.25 \text{ rad}$ . Cette

incertitude est induite d'une part par la nature non déterministe de l'algorithme RANSAC, mais aussi évidemment par les incertitudes des mesures lasers dues au capteur, à l'environnement et au mouvement du robot.

### 5.4.3 Conclusion

Nous avons développé la méthode COMPAS, s'appuyant sur l'algorithme RANSAC, capable de déterminer l'orientation d'un robot dans un environnement structuré composé de couloirs rectilignes. Cette méthode ne peut donc être utilisée dans un environnement non-structuré (halle) ou ne présentant pas majoritairement d'orientation rectiligne (angle, croisement). Cette méthode permet cependant d'améliorer significativement l'incertitude d'orientation par rapport aux méthodes ODO ou KIN-ARUCO.

## 5.5 COMPKIN

Puisque la méthode de localisation KIN-ARUCO présente des incertitudes de localisation importantes entre autres liées aux incertitudes de mesure d'orientation, il semble logique de chercher à coupler KIN-ARUCO avec COMPAS pour définir une nouvelle méthode de localisation que nous désignerons par COMPKIN.

Nous ne nous attarderons pas sur le diagramme causes-effets de COMPKIN puisqu'il correspond en fait à la fusion de celui des méthodes KIN-ARUCO et COMPAS. Nous allons très rapidement revenir sur la formulation intervaliste de son erreur avant de présenter son évaluation expérimentale.

### 5.5.1 Phase 3-1 : Modèle d'erreur théorique

Nous allons tout d'abord rappeler la formulation de l'erreur d'orientation donnée par l'Équation 4.6.

$$\theta_{\mathcal{R}_M|\mathcal{R}_R} = \theta_{\mathcal{R}_M|\mathcal{R}_{Aruco}} + \theta_{\mathcal{R}_{Aruco}|\mathcal{R}_{Kinect}} + \theta_{\mathcal{R}_{Kinect}|\mathcal{R}_R}$$

Dans cette équation la méthode COMPAS va directement nous donner une estimation de  $\theta_{\mathcal{R}_M|\mathcal{R}_R}$  avec une plus faible incertitude que celle identifiée avec la méthode KIN-ARUCO. Par transitivité, cela va donc nécessairement améliorer la précision de la localisation définie par l'Équation 4.5, où l'orientation du robot intervient dans  $MT_{\mathcal{R}_R(t)|\mathcal{R}_{Aruco}}^{-1}$

$$\begin{bmatrix} O_{\mathcal{R}_R(t)|\mathcal{R}_M} \end{bmatrix} = MT_{\mathcal{R}_M|\mathcal{R}_{Aruco}} \bullet MT_{\mathcal{R}_R(t)|\mathcal{R}_{Aruco}}^{-1} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

Le prochain paragraphe va nous permettre d'évaluer l'impact de COMPAS sur KIN-ARUCO.

### 5.5.2 Phases 3-3 et 4 : Évaluation expérimentale et conclusion

Pour procéder à l'évaluation expérimentale de la méthode COMPKIN nous avons mis en œuvre le même processus expérimental que pour la méthode KIN-ARUCO (5.3.3). La Figure 5.12 présente les résultats expérimentaux obtenus à la fois pour la méthode COMPKIN et pour la méthode KIN-ARUCO.

On constate facilement que la diminution de l'incertitude d'orientation engendrée par l'utilisation de COMPAS impacte aussi clairement la précision de localisation et qu'évidemment les

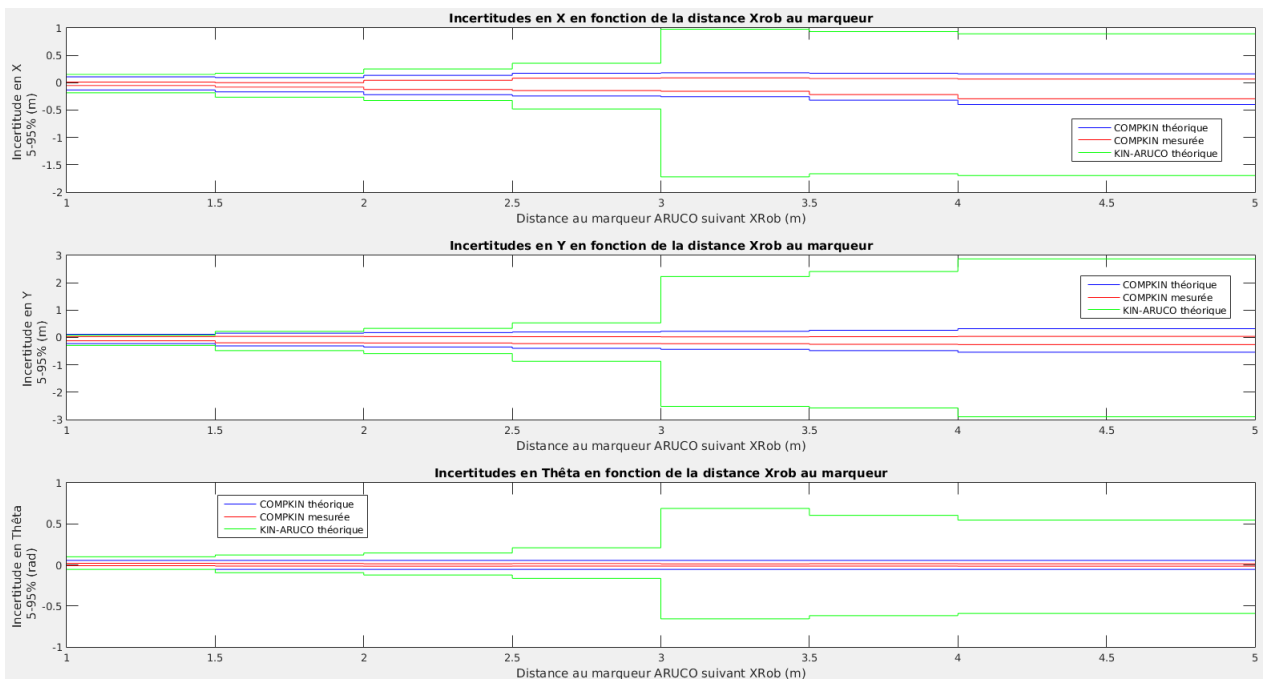


FIGURE 5.12 – Comparaison des méthodes KIN-ARUCO et COMPKIN

relevés expérimentaux corroborent le modèle d'erreur théorique.

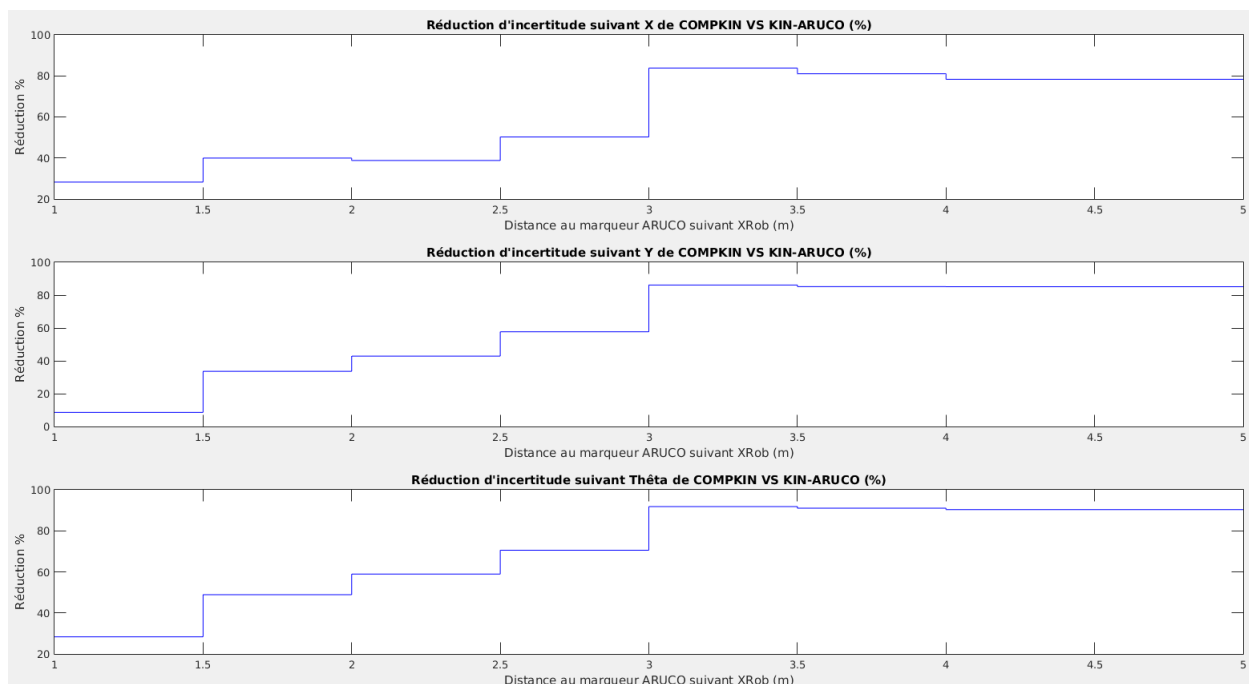


FIGURE 5.13 – Gain de COMPKIN versus KIN-ARUCO

La Figure 5.13 représente le gain obtenu, en pourcentage, grâce à l'utilisation de COMPAS, par rapport à la méthode KIN-ARUCO seule, pour différentes distances au QR-Code dans le repère Robot. Évidemment l'amélioration est d'autant plus importante que la distance au QR-Code est grande puisque l'erreur d'orientation est plus faible pour de petites distances. Par ailleurs, l'emploi de COMPAS nous permet d'étendre de 3 m à 5 m la zone d'utilisation maximale de la Kinect, ce qui constitue un gain substantiel d'espace d'utilisation de COMPKIN par rapport à KIN-ARUCO. De plus, COMPAS rendant la détection des QR-Codes moins dépendante

dante de l'estimation d'orientation du robot par Aruco, il est alors possible avec la méthode COMPKIN, d'envisager d'utiliser des marqueurs présentant un angle d'incidence important par rapport au robot.

Les méthodes KIN-ARUCO et COMPOKIN présentent l'inconvénient de ne pouvoir être utilisées que très localement, au sein de notre mission, au voisinage des marqueurs. Pour être employées plus largement il faut donc qu'elles soient complétées par une méthode de navigation à l'estime et donc, dans notre cas par le recours aux informations odométriques.

## 5.6 KINODO - COMPODOKIN

Cette partie fait plutôt référence à une composition qu'à un couplage de méthodes de localisation. L'adjonction de l'odométrie à KIN-ARUCO va engendrer KINODO, et COMPODOKIN à partir de la méthode COMPKIN.

Là encore nous ne nous attarderons pas sur les diagrammes causes-effets de ces nouvelles méthodes de localisation puisque, comme précédemment, ils résultent de la juxtaposition de ceux des méthodes qui les composent.

### 5.6.1 Phase 3-1 : Modèle d'erreur théorique

Nous allons donc composer à la fois une méthode de recalage global avec KIN-ARUCO ou COMPKIN, avec l'utilisation du modèle odométrique ODO de navigation à l'estime. Pour obtenir le modèle théorique final nous allons donc nous appuyer sur l'équation 4.9 établie dans le chapitre précédent.

$$\begin{bmatrix} O_{\mathcal{R}}(t)|_{\mathcal{R}_M} \end{bmatrix} = \begin{bmatrix} MT_{\mathcal{R}_M|\mathcal{R}_{R(t_r)}} \end{bmatrix} \bullet \begin{bmatrix} O_{\mathcal{R}}(t)|_{\mathcal{R}_{R(t_r)}} \end{bmatrix}$$

D'une part, dans cette équation,  $\begin{bmatrix} O_{\mathcal{R}}(t)|_{\mathcal{R}_{R(t_r)}} \end{bmatrix}$  est obtenu à partir du modèle de dérive odométrique et correspond à la contribution de la méthode ODO.

D'autre part les estimations KIN-ARUCO et COMPKIN nous permettent de construire la partie  $\begin{bmatrix} MT_{\mathcal{R}_M|\mathcal{R}_{R(t_r)}} \end{bmatrix}$  dans l'équation précédente que l'on peut détailler par la matrice de transformation homogène suivante :

$$\begin{aligned} \begin{bmatrix} MT_{\mathcal{R}_M|\mathcal{R}_{R(t_r)}} \end{bmatrix} &= \begin{bmatrix} MT_{\mathcal{R}_M|\mathcal{R}_{QR}} \end{bmatrix} \bullet \begin{bmatrix} MT_{\mathcal{R}_{R(t_r)}|\mathcal{R}_{QR}} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \cos([\theta_{QR}|\mathcal{R}_M]) & -\sin([\theta_{QR}|\mathcal{R}_M]) & [X_{QR}|\mathcal{R}_M] \\ \sin([\theta_{QR}|\mathcal{R}_M]) & \cos([\theta_{QR}|\mathcal{R}_M]) & [Y_{QR}|\mathcal{R}_M] \\ 0 & 0 & 1 \end{bmatrix} \bullet \\ &\quad \begin{bmatrix} \cos([\theta_{QR}|\mathcal{R}_{R(t_r)}]) & -\sin([\theta_{QR}|\mathcal{R}_{R(t_r)}]) & [X_{QR}|\mathcal{R}_{R(t_r)}] \\ \sin([\theta_{QR}|\mathcal{R}_{R(t_r)}]) & \cos([\theta_{QR}|\mathcal{R}_{R(t_r)}]) & [Y_{QR}|\mathcal{R}_{R(t_r)}] \\ 0 & 0 & 1 \end{bmatrix}^{-1} \end{aligned} \quad (5.24)$$

où les intervalles exprimés dans  $\mathcal{R}_M$  correspondent à nos incertitudes de connaissances et ceux exprimés dans  $\mathcal{R}_{R(t_r)}$  à nos incertitudes de mesures et de transformation vers le repère robot depuis le repère capteur.

A l'aspect purement localisation s'ajoute l'incertitude d'orientation qui est toujours classiquement obtenue par la forme intervalliste de l'Equation 4.6.

$$Rot_{\mathcal{R}_M|\mathcal{R}_R} = Rot_{\mathcal{R}_M|\mathcal{R}_{Aruco}} + Rot_{\mathcal{R}_{Aruco}|\mathcal{R}_{Kinect}} + Rot_{\mathcal{R}_{Kinect}|\mathcal{R}_R}$$

Le prochain paragraphe va nous permettre d'évaluer en simulation le comportement des méthodes KINODO et COMPODOKIN sur un environnement disposant de marqueurs Aruco.

### 5.6.2 Phases 3-2 et 4 : Évaluation en simulation et conclusion

Pour évaluer le comportement de l'incertitude de pose nous avons effectué des simulations où le robot réalise une trajectoire rectiligne de 15 m et croise 3 marqueurs Aruco placés respectivement à 4, 12 et 13 m de son point de départ. Les Figures 5.14 et 5.15 présentent les profils d'incertitude prédits par nos modèles. La position des marqueurs est repérée par une flèche. On constate bien évidemment que l'incertitude va dépendre de la distance parcourue à l'estime entre 2 QR-Codes. Cela permet donc de limiter la dérive odométrique.

Nous choisissons dans la suite du document de systématiquement écraser l'estimation de pose odométrique par celle fournie par les QR-Codes dès que celle-ci est utilisable. Cela transparaît au travers des 2 simulations proposées ci-après. Pour la méthode KINODO on peut observer que nous partons d'une incertitude nulle puisque l'on commence par se déplacer à l'estime avant que les informations QR-Code ne puissent nous permettre de nous recalculer. Pour la méthode COMPODOKIN, puisque l'utilisation du COMPAS permet d'étendre la plage d'utilisation des marqueurs Aruco, ces derniers peuvent être détectés dès le début de la simulation ce qui induit, contrairement au cas précédent, une incertitude initiale non nulle.

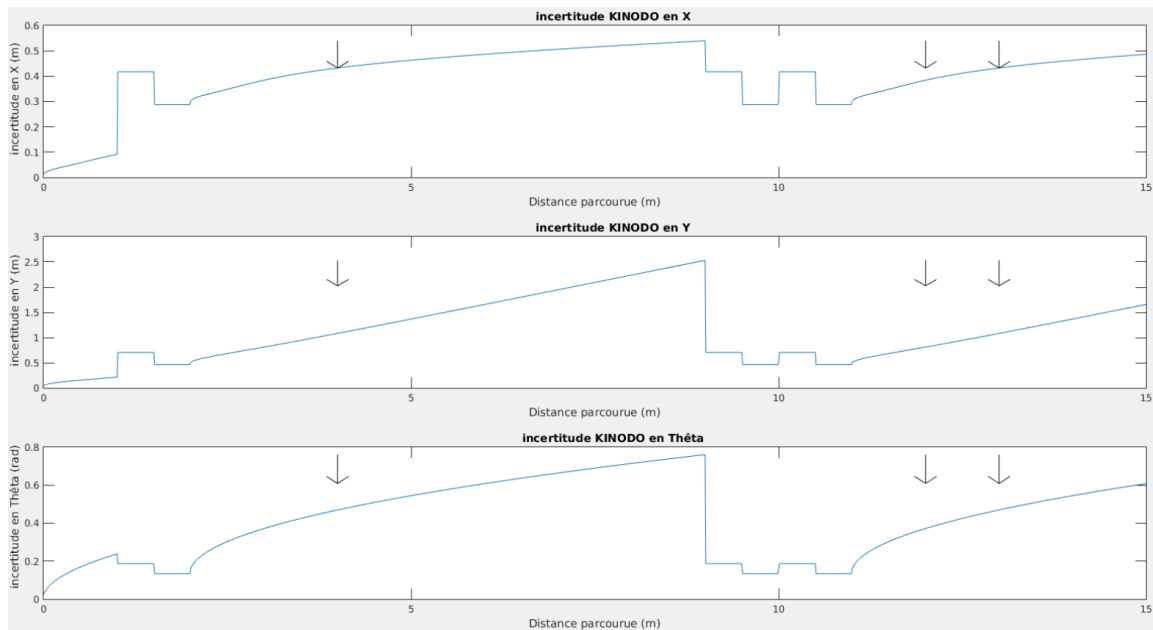


FIGURE 5.14 – KINODO : Simulation de l'évolution de l'incertitude de pose

Si l'on compare les performances de KINODO et COMPODOKIN on constate évidemment que cette dernière permet de se recalculer en étant plus éloigné d'un QR-Code. Cela réduit la distance parcourue à l'estime, et donc l'incertitude maximale atteinte. Cela confirme naturellement que l'équipement préalable de l'environnement aura une influence notable sur la performance de localisation que l'on sera à même d'atteindre. Plus les QR-Codes sont espacés, plus la distance à parcourir à l'odométrie sera importante, cela impactera fortement notre garantie de localisation en raison du modèle très conservatif adopté pour la navigation à l'estime.

N'oublions pas cependant que ces simulations supposent que l'on est toujours capable de détecter les marqueurs ARUCO disposés dans l'environnement s'ils se trouvent excentrés d'au

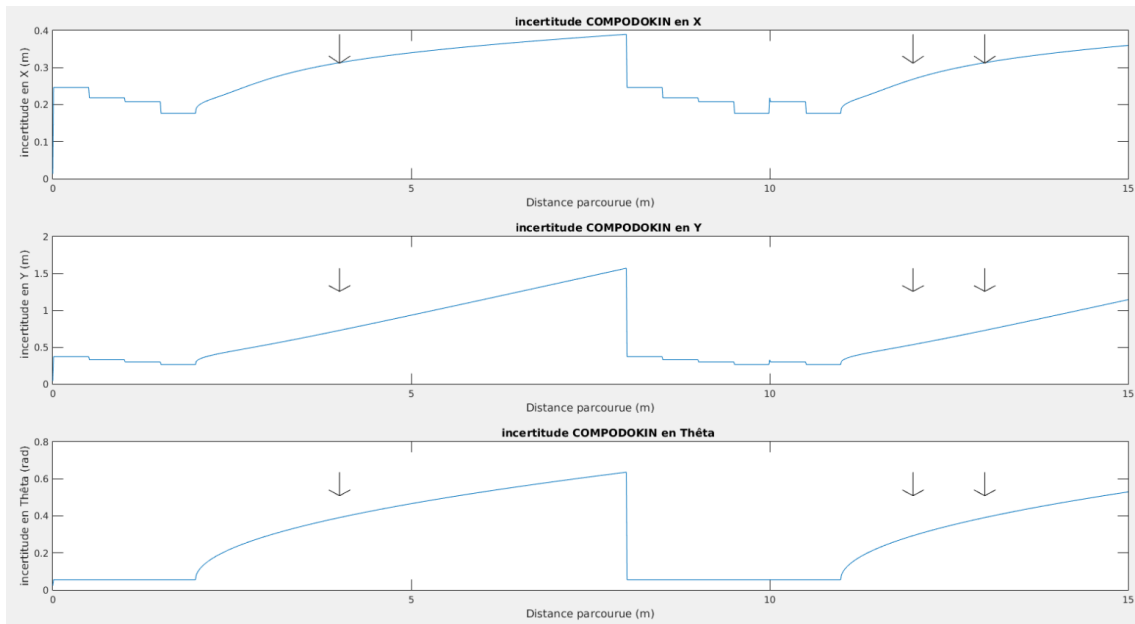


FIGURE 5.15 – COMPODOKIN : Simulation de l'évolution de l'incertitude de pose

plus 1m et à une distance comprise entre 2m et 5m. Si malheureusement un QR-Code n'est pas détecté alors le robot devra se déplacer plus longuement à l'odométrie avant de se recalier. Cela conduira potentiellement à violer les contraintes de localisation imposées.

Pour garantir la détection d'un amers il faut utiliser ceux qui sont placés au plafond et donc faire appelle à la Kinect 2.

## 5.7 KIN2ODO : Localisation extéroceptive par amers artificiels horizontaux

L'étude de la Kinect 2 en tant que telle a déjà été effectuée dans le chapitre précédent. Son diagramme Causes-Effets est bien évidemment similaire à celui de la Kinect 1. Son modèle d'incertitude d'estimation de localisation a été établi expérimentalement au chapitre précédent lors de sa caractérisation dynamique. Il correspond simplement à un pavé d'incertitude de pose. Nous ne nous intéresserons donc pas plus longuement à l'étude de la Kinect 2 mais nous nous focaliserons ici sur l'impact de son utilisation pour recalier périodiquement la dérive odométrique si bien évidemment des QR-Codes sont présents au plafond. Cette méthode de localisation sera dénommée KIN2ODO.

La Figure 5.16 présente, en simulation, l'évolution théorique du profil d'incertitude de localisation de pose. Les QR-Codes sont représentés par des flèches et positionnés aux mêmes endroits (4, 12 et 13 m) que pour l'étude des méthodes KINODO et COMPODOKIN réalisées précédemment.

On peut constater que la méthode COMPODOKIN permet évidemment une détection plus lointaine des QR-Codes puisque KIN2ODO ne les détecte que 40 cm en avant. Cependant l'incertitude initiale de position étant nettement plus faible pour KIN2ODO l'incertitude finale observée après une navigation à l'estime est moindre que pour COMPODOKIN. Donc pour une même incertitude finale la distance parcourue entre 2 recalages va être supérieure pour COMPODOKIN à KIN2ODO car le QR-Code sera détecté de plus loin même si c'est avec une incertitude moins bonne. Malheureusement, cela suppose que l'on peut garantir de toujours



détecter un QR-Code entre 2 et 5 m. En revanche, KIN2ODO garantit de voir les QR-Codes placés au plafond et donc, en les positionnant judicieusement, elle est capable de garantir la performance de pose.

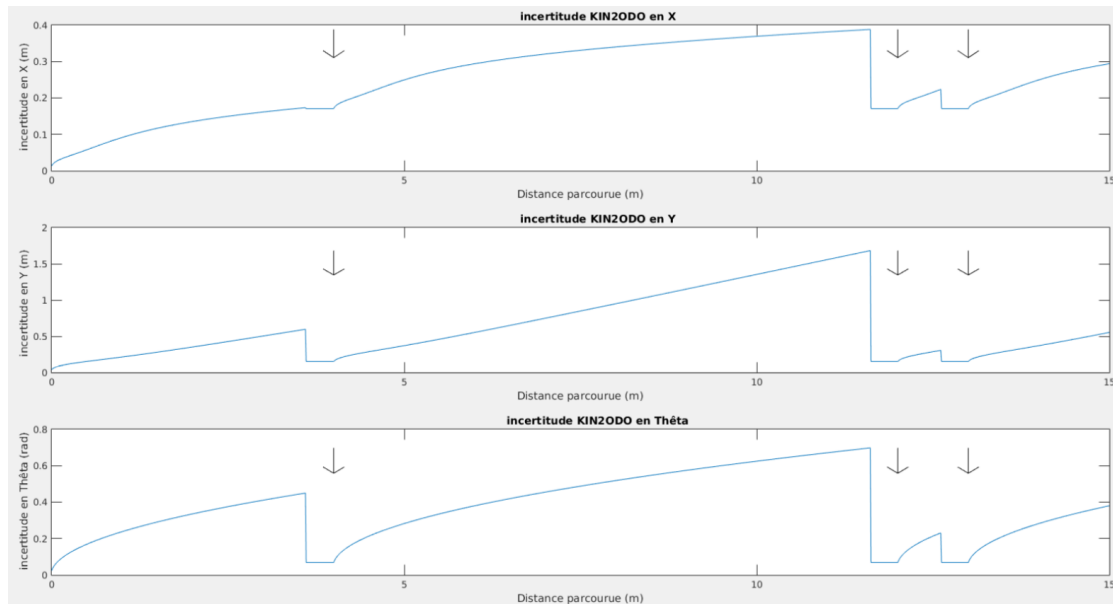


FIGURE 5.16 – KIN2ODO : Simulation de l'évolution de l'incertitude de pose

Nous venons d'établir un ensemble de modèles prédictifs associés à des méthodes locales d'estimation de pose. Il reste cependant pertinent de disposer de méthodes de localisation absolue à même de positionner le robot sans avoir d'information préalable sur sa localisation. C'est pourquoi nous allons maintenant passer à la présentation et à l'évaluation de LZA, une nouvelle méthode de localisation globale.

## 5.8 LZA : Une nouvelle méthode de localisation globale

Cette partie va présenter une nouvelle méthode de localisation globale développée au sein de l'équipe EXPLORE, la méthode LZA. Nous commencerons dans un premier temps par la décrire avant d'analyser son incertitude au travers de l'étude du diagramme d'Ishikawa associé. Puis son modèle d'erreur de principe sera confronté aux résultats obtenus par simulation et expérimentalement. Les résultats obtenus nous conduiront à l'enrichir localement par l'ajout d'informations proprioceptives (odométrie) puis extéroceptives (Compas) pour obtenir le modèle d'incertitude que nous avons évalué expérimentalement. Commençons donc par présenter les principes de cette nouvelle méthode de localisation globale.

### 5.8.1 Description de la méthode LZA

Cette méthode de localisation globale a été initialement développée au sein de l'équipe EXPLORE par le Professeur Zapata [237]. On peut en trouver une vidéo de présentation sur notre chaîne YouTube [56]. Nos travaux ont permis de la confronter aux résultats expérimentaux, de l'évaluer et de chercher à compenser les limitations observées.

Cette méthode cherche donc à déterminer une estimation de la pose  $\hat{\xi} = [\hat{x} \ \hat{y} \ \hat{\theta}]^T$  d'un robot évoluant dans un environnement connu. Il dispose d'un ensemble  $N_s$  de capteurs extéroceptifs lui permettant d'obtenir une signature à  $360^\circ$ , avec un pas régulier, des obstacles distants. Sans perdre en généralité, pour illustrer notre propos, nous considérerons dans ce paragraphe que

la plateforme évolue dans l'environnement de la Figure 5.17 et qu'elle dispose d'un ensemble de 57 capteurs, régulièrement répartis autour du robot, ayant une plage de fonctionnement de 6 m. Ces capteurs sont donc numérotés de 1 à 57, dans le sens horaire, le capteur de référence étant aligné avec l'axe  $x$  de la carte à  $\pi$ . Chacune des mesures capteurs peut par exemple être associée à une unique information télémétrique telles que l'impact d'un faisceau laser, ou à une valeur sonar.

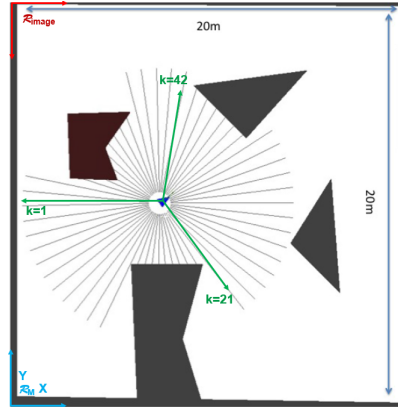


FIGURE 5.17 – Environnement du robot - obstacles et mesures capteurs

Nous allons maintenant détailler les principales phases de la méthode LZA avant d'en présenter les algorithmes simplifiés et de conclure en résumant ses principales propriétés mais aussi certaines de ses limitations.

### 5.8.1.1 Phases de mise en œuvre

**A Phase d'apprentissage (hors ligne)** Durant la phase d'apprentissage, effectuée en simulation, le robot, équipé de ses  $N_s$  capteurs balaie complètement l'environnement, avec une orientation fixe  $\theta = 0^\circ$  en se déplaçant discrètement au centre des cases d'une grille. Les dimensions de cette dernière,  $L_{Max}$  et  $C_{Max}$ , représentent respectivement le nombre de lignes et de colonnes, qui sont choisies en fonction de la résolution désirée.

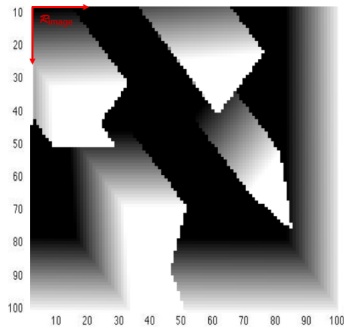
$N_s$  pseudo-images pixélisées de l'environnement, de taille  $L_{Max} * C_{Max}$ , sont construites. Pour un capteur donné, chaque pixel d'une image correspond à ce que mesure ce dernier, depuis chaque position centrale d'une case. Il est donc possible de définir une matrice de transformation homogène  $M$  entre les coordonnées d'un pixel  $(L, C)$  considéré au sein de cette grille et la position "réelle"  $(x, y)$  du robot au sein de l'environnement selon l'Équation 5.25.

$$[L \ C \ 1]^T = M \cdot [x \ y \ 1]^T \quad (5.25)$$

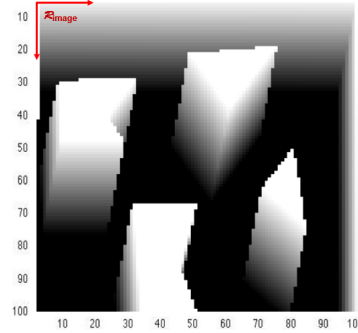
L'intensité  $I(L, C, k)$  du pixel  $(L, C)$  d'une image est choisie de façon à refléter la distance perçue  $D_s(L, C, k)$  par le capteur de rang  $k$ , de plage de mesure  $R$ , au point de coordonnées  $(x, y)$  de la grille selon l'Équation normalisée 5.26.

$$I(L, C, k) = 1 - \frac{D_s(L, C, k)}{R} \quad (5.26)$$

Par exemple supposons que l'environnement de la Figure 5.17 soit recouvert d'une grille composée de cellules de 20 cm de côté. L'environnement sera alors appréhendé au travers de  $N_s$  (57) pseudo-images de  $100 * 100$  pixels en reportant pour chacun d'entre eux l'intensité mesurée pour un capteur donné. Les figures suivantes représentent la pseudo-image obtenue pour les capteurs 21 et 42.



a) LZA : Pseudo-image pour  $k = 21$



b) LZA : Pseudo-image pour  $k = 42$

FIGURE 5.18 – Méthode LZA : Exemples de pseudo-images

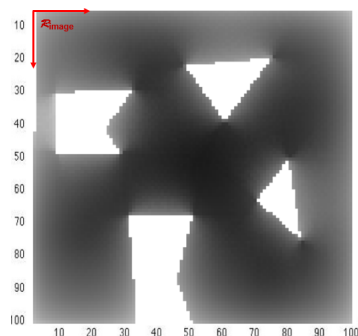
Plus les zones sont claires, plus le capteur est proche (voir à l'intérieur ( $D_s = 0$ ), nous sommes en simulation) d'un obstacle. A contrario, plus les zones sont sombres plus le capteur est éloigné d'un obstacle. Les zones noires indiquent que le capteur est trop loin des obstacles pour les détecter.

Maintenant que l'environnement a été appris et mémorisé, pour chacun des capteurs, sous la forme d'une pseudo-image, la phase de transformation peut commencer.

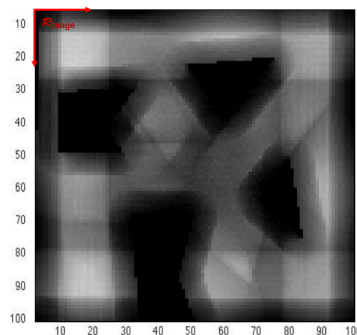
**B Phase de transformation (hors ligne)** De façon à pouvoir déterminer la pose inconnue d'un robot, les pseudo-images obtenues doivent dans un premier temps être transformées de façon à ce que le résultat soit indépendant de l'orientation  $\theta$  inconnue du robot. Les pseudo-images suivantes sont alors construites :

- **Pseudo-Image Énergétique (PIE)** : Cette image  $PI_1$  est obtenue en sommant les  $N_s$  pseudo-images puis en normalisant le résultat.
- **Pseudo-Image Pourcentage (PIPQ%)** : Cette image  $PI_2$  est obtenue en comptant et normalisant le nombre de capteurs d'intensité  $I(:, :, k)$  comprise entre une borne minimale  $P$ , et une borne maximale  $Q$ .

Les figures suivantes, qui sont indépendantes de l'orientation du robot, et où les axes représentent les coordonnées des pixels, montrent les pseudo-images PIE et PIPQ% obtenues pour l'environnement de la Figure 5.17. Pour chaque pixel on dispose donc de 2 signatures spécifiques prenant en compte l'ensemble des données capteurs sélectionnés, et pouvant être associées à une localisation au sein de la grille et donc de l'environnement.



a) LZA : Pseudo-image énergétique



b) LZA : Pseudo-image PQ% pourcentage pour  $P=0.25$  et  $Q=0.75$

FIGURE 5.19 – Méthode LZA : Pseudo-images obtenues à l'issue de la phase de transformation

Les phases hors-ligne étant achevées, les phases en-ligne de localisation peuvent maintenant être entamées.

**C Phase de localisation en (x,y) (en ligne)** Le robot réel est à présent plongé à une position inconnue dans l'environnement à  $t=0$  (*kidnapping*). A partir de ses mesures capteurs  $D_r(k)$  il est alors possible, en utilisant l'Équation 5.26, de calculer l'intensité mesurée depuis sa position courante (Équation 5.27). Dans cette équation, les coordonnées  $L$  et  $C$  du pixel sont maintenant inconnues puisque dépendantes de la position réelle du robot et non d'une position connue simulée.

$$I(L, C, k) = 1 - \frac{D_r(L, C, k)}{R} \quad (5.27)$$

Pour trouver la localisation  $(x,y)$  il faut donc inverser cette équation et donc trouver l'ensemble  $X$  des cases candidates vérifiant  $X(L, C) = \{I(L, C, k) \mid 1 \leq k \leq N_s\}$

Pour parvenir à identifier les cases candidates nous allons calculer en ligne, pour la localisation courante du robot, le coefficient  $CO_1$  correspondant à l'énergie de sa position ainsi que le facteur  $CO_2$  correspondant à  $PQ\%$  (Équation 5.28).

$$CO_1 = \frac{1}{N_s} \sum_{n=1}^{N_s} I(L, C, k) \quad CO_2 = \frac{1}{N_s} \text{card}(p \leq I(L, C, k) \leq q) \mid 1 \leq k \leq N_s \quad (5.28)$$

Une fois ces valeurs déterminées une pseudo-image binaire  $IB_r(t)$  vérifiant 5.29 et identifiant les cases/pixels candidats, sans considérer l'orientation du robot, peut être produite pour un seuillage  $\Xi_n$  donné.

$$IB_r(t) = \bigcap_{n=1}^2 |PI_n - CO_n| \leq \Xi_n \quad (5.29)$$

La Figure 5.20 présente l'image binaire obtenue pour l'exemple traité lorsque le robot est positionné dans l'environnement aux coordonnées  $(x = 10.7m, y = 11.6m)$  cerclées en rouge dans la figure.

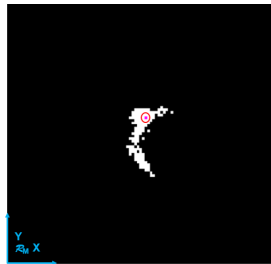


FIGURE 5.20 – Image binaire obtenue pour  $\Xi_n = 0.05 \forall n$

Les pixels candidats  $(L, C)$  peuvent alors être transposés dans l'espace cartésien à partir de l'équation 5.25 en écrivant :

$$[\hat{x} \ \hat{y} \ 1]^T = M^{-1} \cdot [L \ C \ 1]^T \quad (5.30)$$

Puisque plusieurs positions restent candidates nous allons maintenant les départager en cherchant à déterminer l'orientation du robot.

**D Phase de localisation en  $\theta$  (en ligne)** Pour chacun des pixels candidats nous traçons le profil simulé  $D_s(L, C, :)$  et nous le comparons à celui obtenu à partir des informations réelles. Comme le montre la Figure 5.21 ces deux profils présentent une grande similitude d'allure mais sont décalés puisque le premier a été obtenu pour  $\theta = 0$ , qui a peu de chance de correspondre à l'orientation réelle  $\hat{\theta}$  (ici  $45.8^\circ$ ) du robot. Dans cette figure le profil simulé ne présente pas de bruit de mesure ce qui n'est pas le cas du profil réel. Compte tenu du fait que le nombre de pixels candidats peut être relativement important, la sélection du pixel correspondant à la localisation du robot sera particulièrement sensible, surtout dans un environnement non discriminant, au bruit des informations.

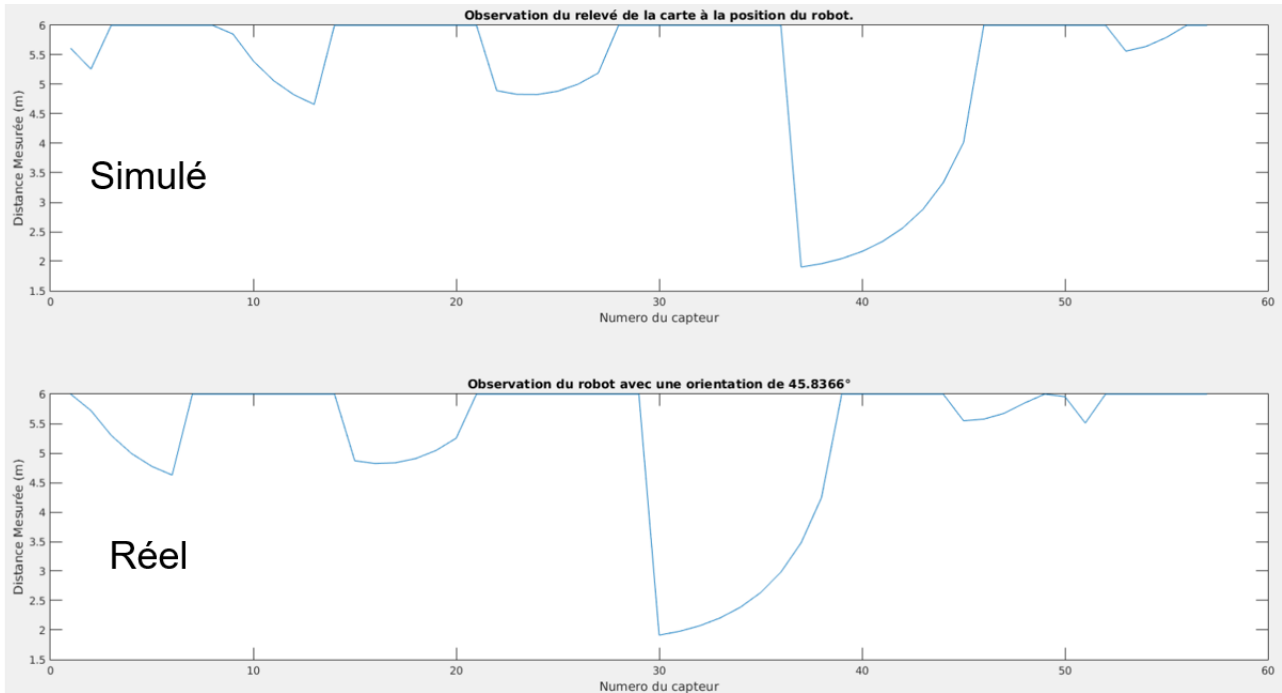


FIGURE 5.21 – Les profils  $D_s$  et  $D_r$  du robot simulé et réel

Puisque ces 2 profils sont corrélés, l'abscisse du maximum de la fonction de corrélation permet donc de déterminer le décalage angulaire entre les 2 profils et donc le cap du robot réel. Le  $\hat{\theta}$  choisi correspond donc à celui qui maximise la corrélation de  $D_r(\cdot)$  en fonction de  $D_s(L, C, :)$ . Cela permet donc pour finir d'obtenir une estimation de la pose  $\hat{P} = (\hat{x}, \hat{y}, \hat{\theta})^T$  du robot.

La mise en œuvre de la méthode LZA repose sur des algorithmes permettant de comparer efficacement les signatures courantes déduites des informations extéroceptives réelles, à celles obtenues au préalable par simulation. Nous allons maintenant les présenter.

### 5.8.1.2 Algorithmes

Conformément à la méthodologie détaillée dans le paragraphe précédent, la méthode LZA peut se décomposer en 2 algorithmes distincts. Le premier, off-line correspond à la phase d'apprentissage (Algorithme 4), le second, on-line, à celle de localisation (Algorithme 5).

L'algorithme d'apprentissage considère que l'environnement est décrit comme un ensemble de polygones. Il déplace le robot au centre de chacune des cases de la grille et pour chacune d'elles, calcule la transformation du repère cartésien au repère image, et la valeur de l'intensité associée à chacun des capteurs. Pour terminer il détermine les pseudo-images issues des fonctions de transformation retenues.

---

**Algorithme 4** : LZA : Phase OFF-LINE

---

- 1: **Phase d'apprentissage**
- 2: Description de l'environnement :  
    {polygone}
- 3: Balayage environnement avec  $\theta = 0$
- 4: **for**  $x = x_{\min}$  to  $x_{\max}$  **do**
- 5:     **for**  $y = y_{\min}$  to  $y_{\max}$  **do**
- 6:         Déplacer le robot à  $(x, y)$
- 7:         Calculer  
            $[L \ C \ 1]^T = M [x \ y \ 1]^T$
- 8:         Pour chaque capteur  $k$  calculer  
            $I(L, C, k)$  (Éq. 5.26)
- 9:     **end for**
- 10: **end for**
- 11: \_\_\_\_\_
- 12: **Phase de transformation**
- 13: Calculer les  $N_{OFPI}$  pseudo-images  
     $CO_n$

---



---

**Algorithme 5** : LZA : Phase ON-LINE

---

- 1: **Phase de localisation à  $t$**
- 2: Description carte :  $N_{OFPI}$  pseudo-images  
     $CO_n$
- 3: Profils : Distances  $D_r(\cdot)$
- 4: Pose du robot  $(x \ y \ \theta)$  inconnue
- 5: Calculer les  $N_{OFPI}$  nombres  $CO_n$
- 6: Calculer  
     $IB_r(t) = \bigcap_{n=1}^{N_{OFPI}} \alpha_n |PI_n - CO_n| < \Xi_n$
- 7: Extraire les couples  $(L \ C)$  tels que  
     $I(L, C) = 1$
- 8: **for** chaque couple  $(L \ C)$  **do**
- 9:     Calculer  $(\hat{x} \ \hat{y})$
- 10:     Calculer la corrélation  
         $C_n = D_r(\cdot)$  vs  $D_s(L, C, \cdot)$
- 11:     Déterminer l'ensemble des cases  
        candidates  $\{(\hat{x}_n \ \hat{y}_n \ \hat{\theta}_n \ C_n)\}$
- 12: **end for**
- 13: Calculer  $\hat{\theta}$  qui maximise ce  $C_n$

---

FIGURE 5.22 – LZA : Algorithmes

L'algorithme de localisation qui dispose en entrée des pseudo-images transformées calculées lors de la phase d'apprentissage et des données réelles provenant du robot peut être décomposé en 2 étapes. D'une part l'étape de détermination des cellules candidates de la grille, et d'autre part l'étape de corrélation qui permet d'identifier la cellule candidate et le cap du robot, et donc la pose estimée de ce dernier.

### 5.8.1.3 Conclusion

La méthode LZA présentée est une méthode de localisation globale puisqu'elle ne fait pas appel à des informations proprioceptives telle que l'odométrie. Qui plus est, ne reposant pas sur des mesures de distance mais plutôt sur des signatures locales associées aux cases d'une grille, nous pouvons sans doute la classifier comme étant une méthode basée sur les apparences même si elle n'est pas utilisée avec une carte topologique de l'environnement.

Son efficacité repose sur trois principaux piliers :

- La capacité des algorithmes à générer et traiter rapidement une grande quantité de données. En effet la phase d'apprentissage nécessite d'explorer l'ensemble de l'environnement pour générer les pseudo-images associées à chacune des cases de la grille. Bien évidemment ce processus est grandement impacté par la résolution choisie. Cette phase conduit donc à mémoriser une très grande quantité d'informations qui doivent être stockées en mémoire. Cependant, même si la phase d'apprentissage est longue elle est effectuée hors-ligne.

En ce qui concerne la phase de localisation on doit distinguer les 2 étapes qui la constituent. Celle d'identification des cases candidates peut être très chronophage car pour les sélectionner il faut parcourir une grande quantité de données mémorisées qu'il faut classer judicieusement pour être efficace, en les ordonnant par exemple en énergie. En revanche si le nombre de cases candidates est restreint, l'étape de corrélation reste très rapide.

En résumé l'efficacité de la phase de localisation repose sur une mémorisation adaptée des données. Sa durée, même si elle peut-être non négligeable reste compatible avec la fréquence de relocalisation qui peut être nettement supérieure à celle de la boucle de contrôle.

- De façon à accélérer l'identification des cellules candidates, une fonction de fenêtrage dynamiquement paramétrable a été introduite. Elle permet de limiter spatialement l'espace de recherche des cellules candidates autour de la localisation estimée du robot lorsque celle-ci est connue. En cas d'échec ou de kidnapping, l'environnement dans sa totalité est à nouveau considéré. On verra que cette fonction jouera un rôle important dans les prochains paragraphes.
- Enfin la pertinence des fonctions de transformation est essentielle puisque ce sont-elles qui vont définir les signatures caractéristiques de l'environnement en un point donné. D'autres choix, s'ils restent indépendants de la rotation, pourraient être envisagés.

La méthodologie LZA présente plusieurs limitations. La première évidemment est que cette méthode de localisation nécessite de disposer d'une connaissance préalable de l'environnement ce qui l'exclut de fait des missions d'exploration. La seconde est que puisqu'elle repose sur un apprentissage s'appuyant sur un environnement statique elle est évidemment très sensible à la dynamique de ce dernier. Elle nécessite évidemment de disposer d'un modèle numérique de l'environnement aussi proche que possible tant de la réalité métrique, que de son impact sur les mesures capteurs. Enfin la résolution choisie pour la grille, le nombre de capteurs considérés (et donc la résolution angulaire de la méthode), ainsi que la multiplicité des fonctions de transformation définies ont un fort impact sur la masse de données à générer et à manipuler.

Pour terminer, il est important de souligner que contrairement aux approches Bayésiennes, la localisation d'un robot n'est pas appréciée en termes probabilistes mais de façon totalement déterministe. Cependant sa pertinence doit être confrontée aux données expérimentales. C'est ce que nous allons faire dans les prochains paragraphes en analysant pratiquement l'incertitude de localisation de LZA et de ses différentes révisions.

## 5.8.2 Méthode LZA : Modèle d'incertitude de localisation

Dans cette partie nous allons analyser l'incertitude de localisation de LZA en la confrontant aux relevés expérimentaux. Nous commencerons par construire le diagramme d'Ishikawa correspondant avant de proposer un modèle d'incertitude empirique qui sera ensuite discuté face aux données expérimentales.

### 5.8.2.1 Phase 1 : Diagramme Causes-Effets

L'incertitude de cette méthode de localisation, décrite par le diagramme d'Ishikawa de la Figure 5.23.

Clairement les dimensions retenues pour grille et l'espacement angulaire entre les relevés de mesures vont directement impacter l'incertitude de pose qu'il est possible, idéalement d'atteindre. Bien évidemment, ces dimensions vont directement influencer le temps de calcul pour obtenir une estimation de la pose et par là même la fréquence maximale d'utilisation de LZA. Plus celle-ci sera importante, plus le robot se sera déplacé depuis la prise de mesure. Enfin, le choix des critères (features) retenus pour construire les pseudo-images est aussi très important. En effet plus ceux-ci sont discriminants, moins on aura de cases à étudier pour déterminer la pose du robot.

La tolérance à ces paramètres  $\Xi_n$  va évidemment impacter la qualité du filtrage permettant de déterminer les cases candidates à la corrélation finale. Par ailleurs, la fenêtre de recherche

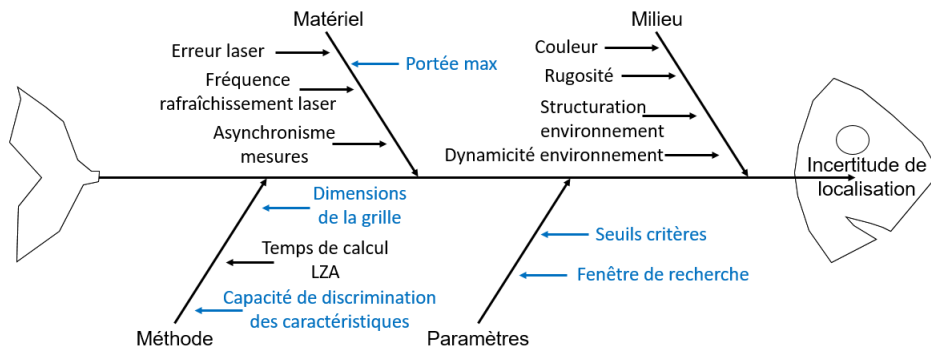


FIGURE 5.23 – Diagramme causes-effets : Erreur de localisation LZA

influence l'incertitude de localisation puisqu'elle permet de limiter la recherche vers une zone où le robot doit normalement se trouver.

L'incertitude de la méthode en elle-même est directement affectée par la dynamique de l'environnement puisque les cartes sont établies dans un environnement statique. Toute variation introduit donc un bruit auquel la méthode sera plus ou moins sensible. Comme beaucoup de méthodes en présence d'un environnement structuré, l'aliasing perceptuel fait que plusieurs cases de la carte peuvent avoir des signatures voisines ce qui les rendra difficilement discernables et donc impactera l'incertitude de localisation. Enfin l'environnement va impacter la qualité de nos mesures en fonction de sa couleur et de son aspect.

L'incertitude est aussi évidemment entachée par la qualité de nos capteurs (portée, erreur), mais aussi par leur fréquence de rafraîchissement. Enfin, le mécanisme d'acquisition de données impacte aussi l'incertitude de localisation de LZA en introduisant une dynamique dans le relevé. D'une part puisque le robot se déplace durant l'acquisition, les données obtenues sont faussées en fonction de la trajectoire suivie. D'autre part, l'acquisition de nos 2 capteurs n'est pas synchrone et donc la fusion en est aussi faussée.

La méthode LZA est donc sujette à de nombreuses sources d'incertitudes. Nous allons cependant, dans un premier temps, établir un modèle d'erreur intervaliste intuitif que nous chercherons à évaluer.

### 5.8.2.2 Phase 3-1 : Modèle d'erreur intervaliste

Pour déterminer le modèle d'erreur de la méthode LZA nous allons nous appuyer sur le Figure 5.24.

Cette figure représente avec un triangle rouge le robot à la position  $t_r$  du dernier recalage LZA. Au moment du recalage, dans un monde complètement discriminant, le robot serait localisé avec certitude au sein d'une unique case. Cependant il existe nécessairement un continuum "de localisation" avec les cases adjacentes. Donc on peut intuitivement s'attendre à ce que le robot soit réellement localisé dans la case identifiée par LZA, ou dans une des cases adjacentes. Sur la Figure 5.24 cela correspond au carré bleu qui relie tous les centres des cases adjacentes à la case choisie par LZA.

Pour notre expérimentation, l'environnement est couvert par une grille de 20 cm de côté et nous disposons de 57 relevés capteurs équirépartis sur  $360^\circ$ , ce qui correspond à un pas angulaire de  $6.32^\circ$ . On peut donc en déduire que l'incertitude théorique de localisation est de  $\pm 20$  cm.

Dans la suite nous allons chercher à vérifier le modèle d'incertitude intervaliste qui vient d'être établi.



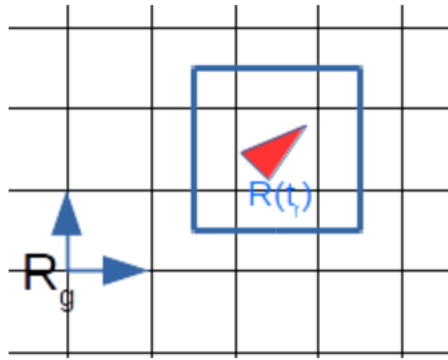


FIGURE 5.24 – LZA : Modèle d’erreur intervalliste

### 5.8.2.3 Phase 3-2 : Modèle d’incertitude : Évaluation statique en simulation

Pour apprécier la validité de notre hypothèse d’incertitude de départ nous avons effectué une simulation massive de LZA sur l’ensemble de l’environnement que nous devons parcourir. Pour se faire nous avons positionné statiquement le robot sur 100 positions aléatoirement distribuées et valides à l’intérieur de chacune des cases de la grille. Pour chacune d’elles, l’algorithme de LZA a été lancé avec une fenêtre de recherche de 4 m de côté centrée autour de la position considérée (localisation locale) ou correspondant à l’environnement dans sa totalité (localisation globale). Une fois le résultat obtenu, nous sommes assurés, ou non, que l’estimation faite par LZA correspondait à l’incertitude que nous avons identifiée (carré bleu de la Figure 5.24).

La Figure 5.25 présente les résultats de cette analyse statistique dans le cas d’une localisation locale où le robot sait approximativement où il se trouve. On constate que globalement LZA semble donner de bons résultats avec un taux de succès supérieur à 80% sur la majorité de l’environnement. Le taux de succès moyen sur l’ensemble de la carte est même de 82.4 %. Par ailleurs, l’analyse statistique fait apparaître que sur la totalité de l’environnement, lorsque LZA se localise, l’erreur gaussienne en X et Y observée présente des écarts-types proches de  $\sigma = 0.28m$  chacune ce qui nous permet d’identifier la taille minimale d’une fenêtre de recherche. On peut remarquer aussi que certaines zones de l’environnement semblent être particulièrement difficiles pour LZA. La méthode ne peut clairement pas être employée dans la halle d’expérimentation, et dans des zones faiblement discriminantes plus localisées de l’environnement. Dans l’approche PANORAMA, LZA ne pourra pas être utilisée dans l’ensemble de ces régions.

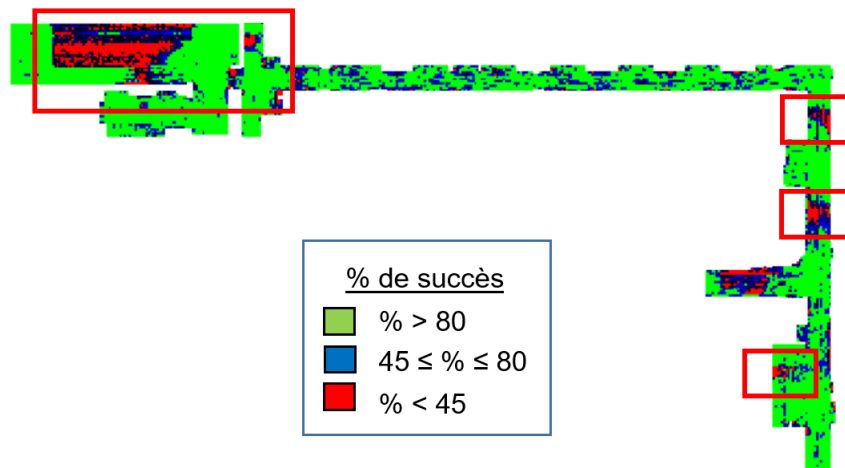


FIGURE 5.25 – LZA (localisation locale) : Évaluation statique et statistique

La Figure 5.26 présente les résultats de cette analyse statistique dans le cas d'une localisation globale où le robot est initialement perdu (*kidnapping*). On constate là encore que LZA semble donner de bons résultats avec un taux de succès supérieur à 80% sur la majorité de l'environnement et un taux moyen de 71.4% inférieur au taux précédent sans être pour autant faible. Cette analyse confirme de façon encore plus discriminante les zones précédentes où la méthode LZA ne pouvait être employée, en y ajoutant tout de même certains passages au sein du couloir 1. En revanche elle démontre l'aptitude de la méthode LZA à adresser une problématique de localisation globale lorsque l'environnement s'y prête.

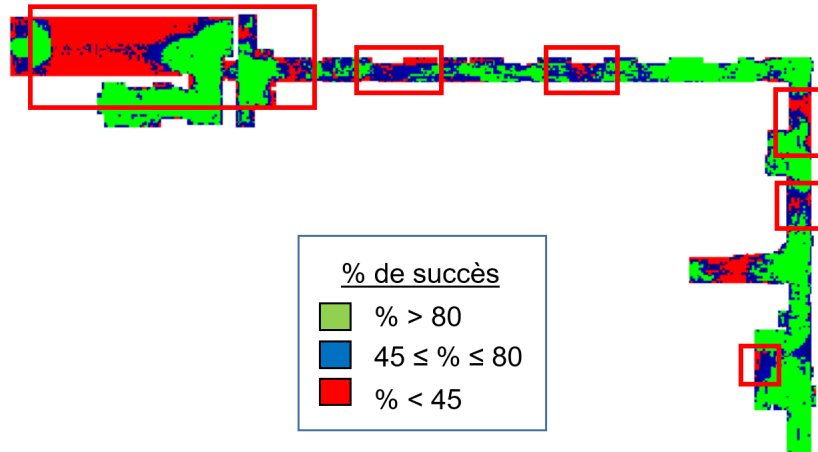


FIGURE 5.26 – LZA (localisation globale) : Évaluation statique et statistique

Ces premiers résultats semblent encourageants et nous permettent de confirmer le bon comportement de LZA tant dans le cadre d'une localisation globale que locale. Dans ce dernier cas nous avons caractérisé une fenêtre minimale de recherche, centrée sur la localisation supposée correcte du robot, de façon à préserver la position réelle du robot à l'intérieur de l'espace de recherche. Cependant, cette analyse est effectuée en statique en l'absence de tout mouvement du robot. Pour prendre en compte des conditions réelles d'utilisation nous allons maintenant évaluer expérimentalement et dynamiquement les performances de localisation de LZA.

#### 5.8.2.4 Phases 3-3 et 4 : Modèle d'erreur : Évaluation dynamique expérimentale

Pour effectuer l'évaluation dynamique expérimentale de la méthode LZA nous avons fait évoluer notre robot sur 25 m, en boucle ouverte, dans la zone expérimentale habituelle du couloir 2. Dans un premier temps, nous avons effectué 8 expérimentations à 0.4 m/s pour lesquelles nous avons enregistré les informations de pose données par l'odométrie et les informations laser recueillies tout au long de ces expériences. Dans un second temps nous avons utilisé la méthode LZA avec les informations laser précédemment recueillies pour déterminer la pose du robot en adoptant une fenêtre de recherche de 4m de côté centrée sur la dernière position estimée par LZA. Pour terminer nous avons comparé ces dernières avec les informations de pose odométriques considérées comme étant la réalité terrain en supposant une dérive limitée.

La Figure 5.27 présente les erreurs de pose mesurées tout au long du couloir 2 pendant ces expériences.

Tout d'abord globalement LZA semble donner la plupart du temps des résultats cohérents. Dans la direction du déplacement (X), on observe une erreur pratiquement centrée à +/- 0.5 m de la réalité terrain. Cependant on identifie clairement 3 zones, repérées par des rectangles rouges, pour lesquelles la méthode de localisation a des problèmes évidents. Ces zones correspondent à des régions à fort aliasing perceptuel.

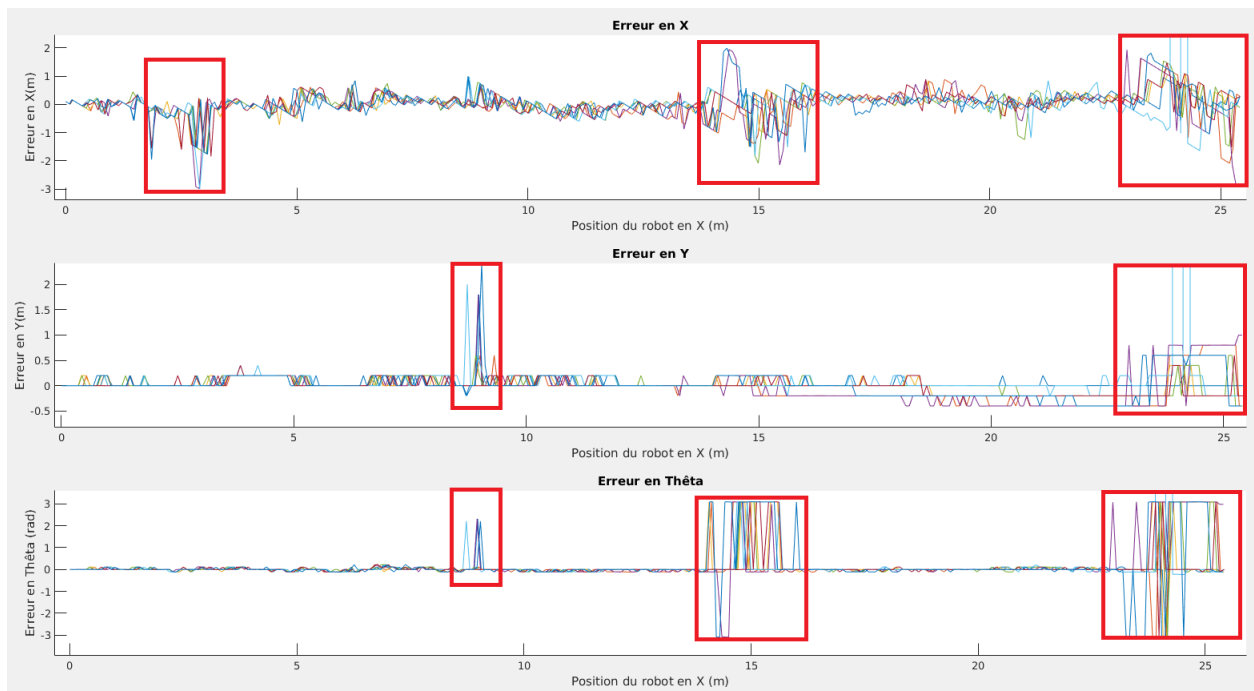


FIGURE 5.27 – LZA : Évaluation dynamique expérimentale

Dans la direction perpendiculaire au déplacement (Y), la même remarque générale peut être énoncée et l'erreur observée ne dépasse pas  $\pm 0.4$  m. On repère aussi des valeurs aberrantes sur 2 zones de l'environnement. La première vers 9 m correspond au passage d'une porte coupe-feu qui est difficilement modélisable dans l'environnement utilisable en apprentissage. La seconde, en fin de mission est aussi liée à l'aliasing qui va perturber, comme nous allons le voir, l'estimation d'orientation. Enfin on peut constater, sur certaines missions, une augmentation de l'erreur en fin de parcours. On peut supposer que cela soit en partie dû à l'accroissement de l'erreur odométrique transversale qui perturbe notre réalité terrain.

Enfin, l'erreur d'orientation reste très faible et comprise entre  $\pm 0.2$  rad conformément au pas angulaire choisi pour la méthode LZA. Malheureusement, il existe encore des zones pour lesquelles on observe une erreur de cap avoisinant  $\pi$  induite par la forte symétrie du couloir. Cela peut effectivement entraîner les fortes erreurs de localisation dont nous avons déjà parlé plus haut. Il faudra donc impérativement chercher à filtrer ces erreurs d'orientation.

Intéressons nous maintenant, non pas à l'erreur de pose, mais dynamiquement à sa qualité. Nous avons cherché à savoir, tout au long d'une mission, si la méthode LZA nous localisait correctement, mal ou était perdue, pour une fenêtre de dimension fixe égale de 4m de côté. Cela a permis d'établir le schéma de la Figure 5.28. Son analyse permet de constater que LZA est rarement perdue mais que la méthode alterne des périodes où elle localise correctement le robot avec d'autres où elle le localise imparfaitement. Cela met en évidence une des limitations de la méthode LZA dans sa forme première. Quand peut-on être certain de la localisation retournée par LZA ? et quand peut-on considérer que LZA donne des résultats erronés ?

Bien que globalement la méthode LZA semble avoir de bonnes performances de localisation en dehors des zones problématiques, les erreurs observées pratiquement dépassent nettement en X et en Y celles que le modèle d'erreur intuitif prévoyait ( $\pm 20$  cm). De plus, il faut rappeler ici que nous nous inscrivons dans une logique de garantie. Il va donc falloir réduire l'impact des artefacts de localisation et rendre LZA plus robuste à l'instabilité de localisation observée. Ces deux questions sont abordées dans les paragraphes suivants.

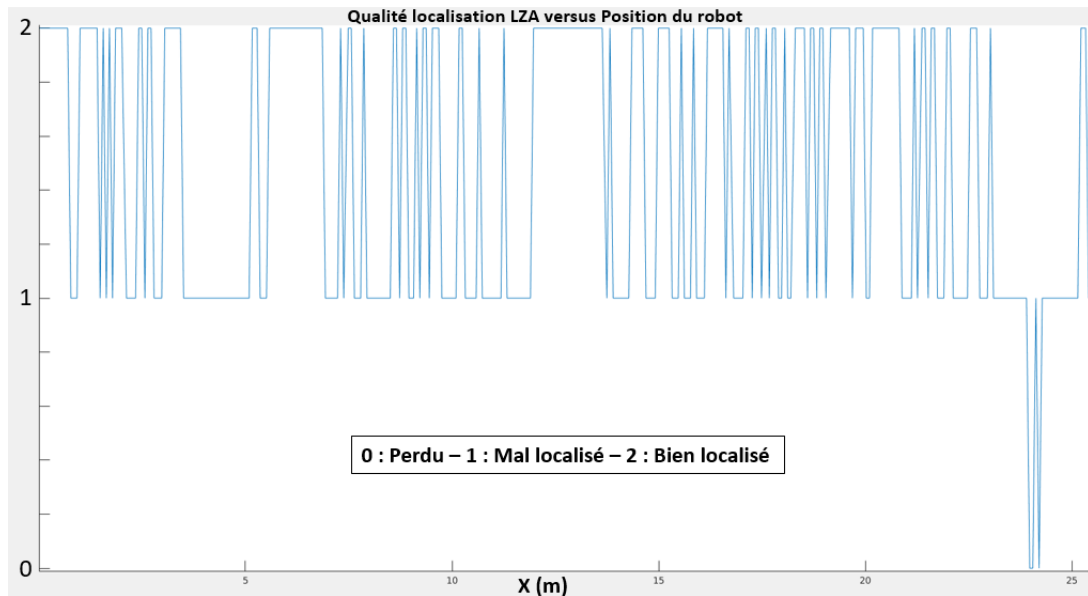


FIGURE 5.28 – LZA : Évaluation de la qualité de localisation

### 5.8.3 Méthode LZA-ODO : Amélioration de la relocalisation de LZA

Dans cette partie nous allons chercher à répondre aux 2 dernières questions posées : Quand peut-on être certain de la localisation retournée par LZA ? et quand peut-on considérer que LZA donne des résultats erronés ?

#### 5.8.3.1 Phase 3-2 : Évaluation en simulation

Les idées suivies pour limiter l'instabilité de la méthode LZA sont d'une part de statuer sur l'état Localisé/Perdu sur un horizon glissant et, d'autre part de contrôler la dimension de la fenêtre de recherche. Dans le cas où le robot serait perdu, celle-ci sera fixée à l'environnement, dans sa totalité. Dans le cas qui nous intéresse ici, où nous adoptons une logique de suivi de posture et où la position du robot est approximativement connue, une fenêtre de 4 m centrée autour de la position supposée du robot est adoptée lors de l'initialisation, ou suivra la dérive odométrique en l'absence de relocalisation par LZA-ODO. Nous avons dénommé cette robustification de LZA, LZA-ODO puisque l'odométrie vient guider l'évolution de la fenêtre de recherche une fois le robot supposé localisé.

Pour répondre aux questions posées plus haut l'algorithme de la Figure 5.29 est proposé. Il fait appel à deux compteurs *LOC\_Valide* et *ECHEC\_Loc* qui évolueront en même temps, mais de façon complémentaire. On distingue 2 seuils pour le compteur *LOC\_Valide*. D'une part *Seuil\_Valide\_P* dans le cas où le robot était perdu ou ne s'est jamais localisé. D'autre part *Seuil\_Valide\_L* lorsque l'algorithme LZA-ODO estime que le robot est localisé. Pour que la localisation renvoyée par LZA-ODO soit considérée comme acceptable et incrémente le compteur correspondant (*LOC\_Valide*) on va s'assurer que cette dernière retourne des informations de pose, avec une orientation cohérente (proche de l'estimation odométrique à  $\delta(\theta)$  près) au sein de la fenêtre de recherche. Si ce n'est pas le cas, l'information de pose renvoyée par LZA sera supposée incohérente et on incrémentera le compteur correspondant (*ECHEC\_Loc*). Pour une situation donnée (Localisé / Perdu) lorsque le compteur *LOC\_Valide* atteint le seuil correspondant, la dernière estimation de LZA-ODO est considérée valide et vient écraser la pose odométrique. A partir de cet instant, la fenêtre de recherche de LZA-ODO est restreinte à  $\pm 0.55 \text{ m}$  ( $2 \sigma$ ) puis suivra l'évolution de l'incertitude odométrique (fonction *Sim\_Odo()*) et tous

les compteurs sont réinitialisés. On attendra que le robot soit éloigné d'une distance  $D_{start}$  de sa dernière localisation pour relancer l'algorithme LZA-ODO de façon à accroître le gradient de signature. Lorsque le compteur  $ECHEC\_Loc$  atteint sa valeur seuil tous les compteurs sont aussi remis à zéro et le robot n'est plus considéré comme localisé.

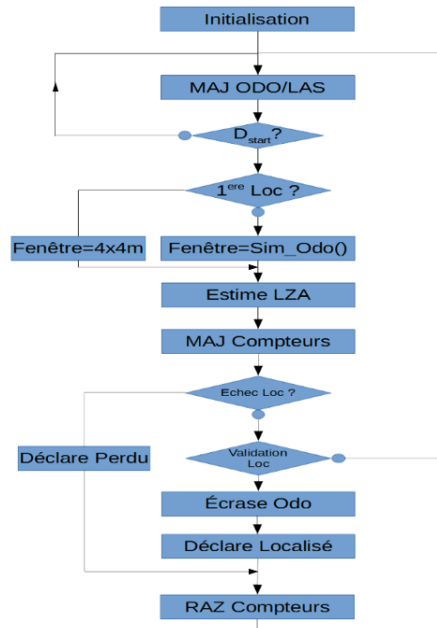


FIGURE 5.29 – LZA : Principe de l'algorithme de filtrage

L'algorithme que nous venons d'exposer présente plusieurs paramètres qui doivent être précisés :

- Le seuil  $Seuil\_Valide\_P$  pour que le robot soit relocalisé à partir d'un état Perdu.
- Le seuil  $Seuil\_Valide\_L$  pour que le robot soit relocalisé à partir d'un état Localisé.
- Le seuil  $Seuil\_ECHEC\_Loc$  faisant passer le robot de l'état Localisé à Perdu.

Pour déterminer les valeurs de ces paramètres nous avons fait appel à une approche génétique évolutionnaire [74] appliquant l'algorithme proposé sur des relevés expérimentaux issus de notre environnement.

Cette démarche permet en plus de déterminer les valeurs d'autres paramètres pertinents :

- Composantes de la matrice de covariance  $\sum_{p\_init}$  associée à la position initiale recalée du robot.
- Distance  $D_{start}$  devant être parcourue par le robot après un recalage avant de relancer LZA-ODO.
- Tolérance d'erreur d'orientation (estimation odométrique / estimation LZA)  $\delta(\theta)$  acceptée.
- Paramètres  $\Xi_n$  de LZA (Équation 5.29).

Cette approche génétique évolutionnaire a été effectuée sur 20 jeux de données représentatifs de notre environnement correspondant aux données brutes lasers et odométriques mémorisées en boucle ouverte lors des déplacements du robot. A partir d'une population initiale aléatoire constituée de 300 individus (jeux de paramètres), près de 250 générations ont été nécessaires avec un taux de renouvellement de 17 % pour déterminer les paramètres conduisant à l'incertitude de pose la plus faible. Leurs valeurs sont présentées dans la Table 5.1.

Paramètres	Valeur
$D_{Start}$	0.74 m
$Seuil\_Valide\_P$	19
$Seuil\_Valide\_L$	14
$Seuil\_ECHEC\_Loc$	19
$\delta(\theta)$	+/- 2.3173 rd
$\sigma_E$	2.58
$\sigma_{PQ\%}$	3.46
$\Sigma_{p\_init}$	[0.1182 0 0; 0 0.1432 0; 0 0 0.1622]

TABLE 5.1 – LZA : Paramètres optimaux issus de l’algorithme génétique

L’algorithme développé pour filtrer les instabilités observées de LZA étant présenté ainsi que la méthode employée pour déterminer les paramètres permettant de le mettre en œuvre, nous allons maintenant évaluer expérimentalement ses performances.

### 5.8.3.2 Phase 4 : Évaluation expérimentale du modèle d’erreur

L’évaluation expérimentale a été mise en œuvre dans la zone habituelle du couloir 2 que nous avons parcourue une douzaine de fois en suivant une trajectoire rectiligne, en boucle fermée, à la vitesse de 0.3 et 0.56 m/s. Les relevés expérimentaux obtenus sont présentés en Figure 5.30. On remarquera que conformément à la Figure 5.25 on exclue par rapport à l’espace d’expérimentation 2 zones où LZA n’est pas éligible. D’une part la zone voisine du laser de réalité terrain où l’expérimentation s’achève, d’autre part la zone sans point de mesure au centre du relevé pour laquelle la localisation est assurée grâce à la Kinect 2.

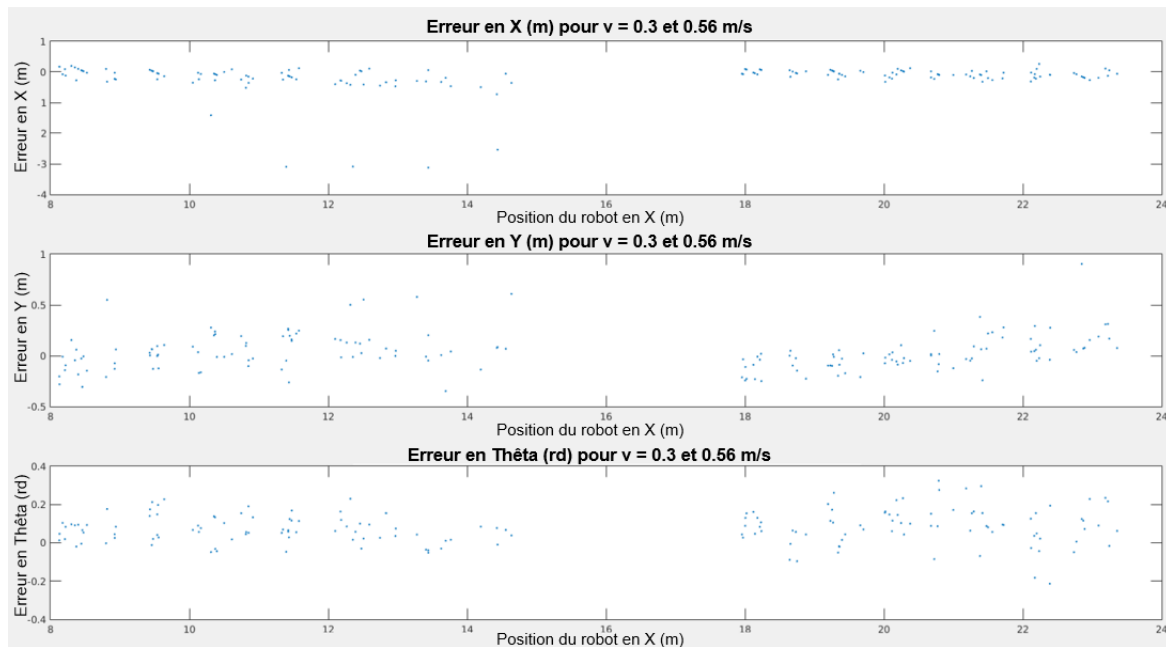


FIGURE 5.30 – LZA-ODO : Relevés expérimentaux

Si l'on retient un intervalle de confiance de 90 % (proche de  $2\sigma$ ) on obtient les pavés d'incertitude suivants :

- Pavé d'incertitude selon l'axe x :  $LZA - ODO([x]) = [-0.3662, 0.10016]$  en m
- Pavé d'incertitude selon l'axe y :  $LZA - ODO([y]) = [-0.2121, 0.30923]$  en m
- Pavé d'incertitude d'orientation :  $LZA - ODO([\theta]) = [-0.043187, 0.22865]$  en rad

Incertitude 9 : Pavés d'incertitude de LZA-ODO (90%)

On constate donc que selon l'axe X l'incertitude est diminuée de pratiquement 60 % par rapport à celle de LZA, mais qu'ici elle n'est pas centrée. Selon l'axe Y, l'amélioration est moindre mais atteint tout de même de près de 40 %. L'incertitude y demeure pratiquement centrée. Enfin pour l'orientation on gagne environ 35 % de précision. Malgré ces améliorations, les limites observées restent légèrement supérieures à celles identifiées par le modèle théorique d'incertitude.

Nous avons choisi de compenser les biais observés et nous utiliserons pratiquement les intervalles suivants pour la méthode LZA-ODO :

- Pavé d'incertitude selon l'axe x :  $LZA - ODO([x]) = [-0.233, 0.233]$  en m
- Pavé d'incertitude selon l'axe y :  $LZA - ODO([y]) = [-0.261, 0.261]$  en m
- Pavé d'incertitude d'orientation :  $LZA - ODO([\theta]) = [-0.136, 0.136]$  en rad

Incertitude 10 : Pavés d'incertitude de LZA-ODO sans biais (90%)

A la vue de ces résultats on peut donc conclure que l'algorithme de robustification proposé ainsi que les paramètres déterminés à l'aide de la démarche génétique évolutionniste permettent une amélioration significative de la précision de localisation obtenue par LZA. Cependant l'estimation d'orientation de LZA reste limitée par le pas retenu lors de la phase d'apprentissage. L'utilisation de la méthode COMPAS préalablement étudiée va nous permettre d'améliorer encore les performances de LZA-ODO.

#### 5.8.4 Méthode LZA-ODO-COMPAS

Lorsqu'elle est possible, l'intégration de COMPAS vient simplement remplacer l'estimation d'orientation de la pose valide qui vient écraser la valeur odométrique. Cela modifie donc très marginalement l'algorithme de filtrage proposé que nous allons maintenant évaluer expérimentalement.

##### 5.8.4.1 Phase 4 : Évaluation expérimentale du modèle d'erreur

Comme précédemment l'évaluation expérimentale a été mise en œuvre dans la zone du couloir 2 que nous avons parcourue quatorze fois en suivant une trajectoire rectiligne, en boucle fermée à la vitesse de 0.3 et 0.56 m/s. Les relevés expérimentaux obtenus sont présentés en Figure 5.31. On remarquera que comment précédemment ce relevé n'est effectué que dans les zones où LZA est éligible, la méthode COMPAS l'étant aussi.

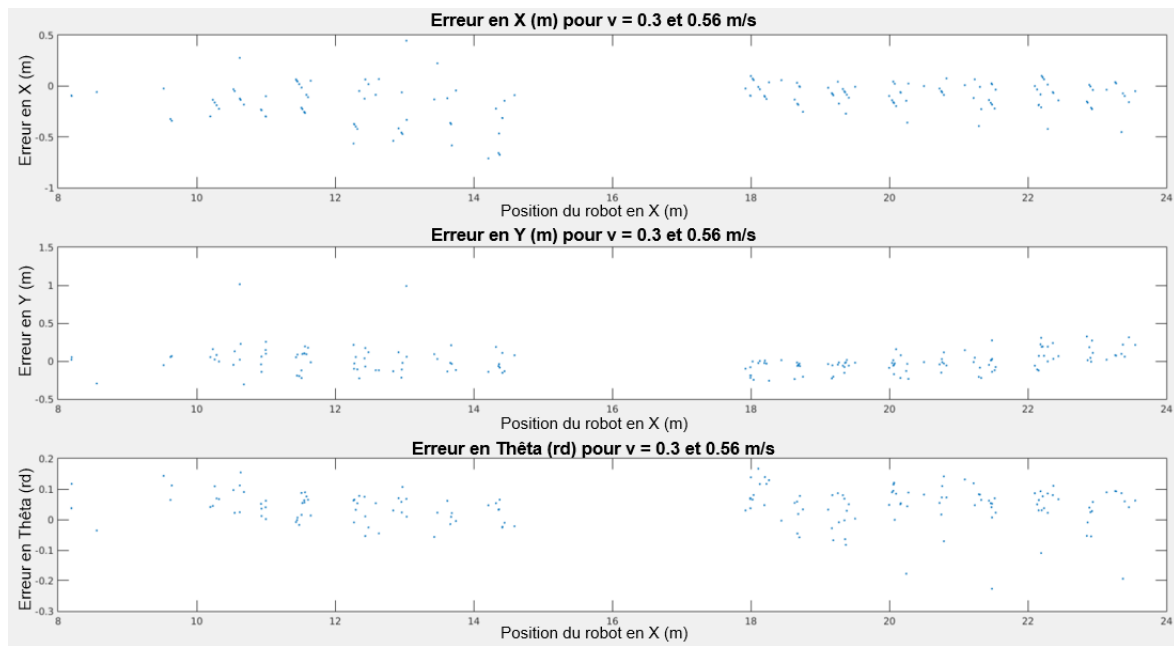


FIGURE 5.31 – LZA-ODO-COMPAS : Relevés expérimentaux

Si l'on retient un intervalle de confiance de 90 % (proche de  $2\sigma$ ) on obtient les pavés d'incertitude suivants :

- Pavé d'incertitude selon l'axe x :  
 $LZA - ODO - COMPAS([x]) = [-0.42181, 0.064962]$  en m
- Pavé d'incertitude selon l'axe y :  
 $LZA - ODO - COMPAS([y]) = [-0.22157, 0.27115]$  en m
- Pavé d'incertitude d'orientation :  
 $LZA - ODO - COMPAS([\theta]) = [-0.067667, 0.13164]$  en rad

Incertitude 11 : Pavés d'incertitude de LZA-ODO-COMPAS (90%)

A la vue de ces nouveaux résultats on constate que les erreurs observées selon les axes X et Y restent sensiblement équivalentes à celles obtenues pour LZA-ODO. Par contre, l'utilisation de COMPAS permet d'améliorer de 26 % la précision d'orientation. Globalement ces mesures vérifient pratiquement le modèle d'incertitude théorique supposé, si ce n'est que l'incertitude en X présente un biais de près de 25 cm en partie imputable aux conditions expérimentales d'obtention (déplacement et environnement orientés principalement selon X).

La compensation des biais observés conduit aux intervalles suivants pour la méthode LZA-ODO-COMPAS :

- Pavé d'incertitude selon l'axe x :  
 $LZA - ODO - COMPAS([x]) = [-0.243, 0.243]$  en m
- Pavé d'incertitude selon l'axe y :  
 $LZA - ODO - COMPAS([y]) = [-0.246, 0.246]$  en m
- Pavé d'incertitude d'orientation :  
 $LZA - ODO - COMPAS([\theta]) = [-0.100, 0.100]$  en rad

Incertitude 12 : Pavés d'incertitude de LZA-ODO-COMPAS sans biais (90%)



### 5.8.5 Conclusion

Nous venons de proposer et d'évaluer une nouvelle méthode de localisation globale LZA, basée sur les apparences, pouvant être employée dans un environnement statique connu couvert par une grille. Elle s'appuie, pour chacune des cases, sur un apprentissage préalable de signatures déduites à partir d'informations extéroceptives équiréparties sur  $360^\circ$ . L'analyse en simulation a pointé les zones de l'environnement où LZA ne pourrait être déployée en raison de l'aliasing perceptuel. L'expérimentation a démontré dans un premier temps une certaine instabilité des valeurs de localisation retournées. L'utilisation conjuguée de LZA et d'un algorithme de filtrage de ses instabilités gérant la dimension de la fenêtre de recherche a permis de stabiliser les errements de LZA. Son comportement expérimental avec ou sans l'adjonction de la méthode COMPAS est alors voisin de celui escompté. Nous disposons donc maintenant d'une méthode globale permettant de recalculer la pose du robot avec une incertitude de pose connue et fixe en tout point de l'environnement où LZA-ODO(-COMPAS) est éligible.

## 5.9 Conclusion

Nous venons d'établir des modèles d'erreur prédictifs pour l'ensemble des méthodes que nous avons à notre disposition sur notre plateforme robotique en nous appuyant sur une démarche rationnelle qui pourrait être utilisée pour toute autre méthode de localisation et fortement ancrée sur la validation expérimentale.

Nous nous sommes tout d'abord intéressés à l'odométrie. C'est une approche de localisation facile à déployer, ne coûtant rien sur le plan énergétique mais qui, étant proprioceptive, a le défaut majeur de dériver avec le déplacement du robot. Elle est cependant incontournable, car en l'absence de toute autre information, elle délivre une bonne estimation de la localisation sur des distances réduites lorsque l'on se déplace à l'estime. Pour ce capteur, sans oublier l'impact des erreurs statiques, un modèle d'incertitude de l'évolution des bornes maximales de l'erreur odométrique de pose a été construit en partant d'un modèle analytique et des relevés expérimentaux effectués.

La méthode de recalage KIN-ARUCO s'appuyant sur la lecture de QR-Codes de type Aruco, a ensuite été évaluée. L'approche intervalliste conservatrice et l'incertitude sur l'orientation observée la rend difficilement utilisable au delà d'une distance de 3m. Cela nous a conduit à proposer une méthode logicielle COMPAS, permettant de déterminer, lorsque cela est possible, avec une bonne précision, l'orientation du robot à partir de l'exploitation des informations extéroceptives issues des lasers. Ainsi le couplage de KIN-ARUCO et de COMPAS a permis de caractériser la méthode COMPKIN améliorant significativement les performances de KIN-ARUCO et permettant d'exploiter les données de la Kinect jusqu'à 5m. Enfin nous avons composé les modèles d'incertitude de recalage avec le modèle de dérive odométrique pour présenter les profils d'incertitudes qui devraient être observés avec les méthodes KINODO et COMPODOKIN. Cependant l'utilisation de la Kinect 1 avec les amers verticaux ne garantit pas la détection de ces derniers. C'est pourquoi nous avons étudié la méthode KIN2ODO qui fait appel à la Kinect 2 pour observer des amers horizontaux placés au plafond avec une garantie de détection, mais sur un espace moindre que COMPODOKIN.

Pour terminer, une nouvelle méthode globale LZA, s'appuyant sur des relevés laser sur  $360^\circ$  a été décrite, évaluée, enrichie et expérimentée permettant de disposer de pavés d'incertitude garantie dans les zones où LZA est éligible.

Il faut tout de même mettre en avant plusieurs points qui nous semblent pertinents concernant la définition et la validation expérimentale de ces modèles prédictifs de garantie de Localisation :

- Ils reposent sur la connaissance d'une réalité terrain dont la précision ne peut être considérée qu'à l'aune de ce qu'il a été possible de mobiliser ou de mettre en place pour l'obtenir.
- L'évaluation expérimentale a nécessité la réalisation d'un nombre très important d'expérimentations.
- Ces travaux confirment évidemment l'impact significatif de l'erreur d'orientation du robot sur la qualité de la localisation.
- Les modèles d'incertitude obtenus s'inscrivant dans une logique intervaliste imposée par la recherche de garantie visée par la méthodologie PANORAMA conduit à la définition d'intervalles pouvant paraître importants. Cela traduit la forte sensibilité des méthodes ensemblistes à la présence de valeurs aberrantes au sein des relevés expérimentaux. De façon à préserver des pavés de dimensions acceptables nous avons éliminé les 5% des valeurs extrémales les plus aberrantes (proche de  $2\sigma$ ) [195]. Quoiqu'il en soit cela souligne que la pose du robot n'est connue à un instant donné qu'avec une précision toute relative.

Tout comme pour les axes de performance Durée et Énergie nous disposons à présent de modèles prédictifs pour l'axe Localisation. Par rapport aux travaux de [112] cette objectivation est une avancée significative qui va impacter le processus d'allocation des ressources logicielles et matérielles de la méthodologie PANORAMA et va nous permettre de nous inscrire dans une logique de maximisation de la marge énergétique.



# Chapitre 6

## VERS UNE GARANTIE DE PERFORMANCE INCLUANT LA LOCALISATION : PRINCIPES DE MISE EN ŒUVRE

Nous disposons à présent, pour chacune des méthodes de localisation déployées sur notre plateforme et leurs différentes implantations en fonction des capteurs mobilisés, de lois prédictives d'évolution de la précision de localisation et de pose. Nous pouvons donc nous y adosser pour intégrer quantitativement l'axe de localisation à la méthodologie PANORAMA qui était à l'origine de notre réflexion. Le présent chapitre a donc pour objectif de présenter les différentes phases de cette intégration ainsi que l'impact que cela va avoir sur la dimension énergétique et sur la génération des plans de mission.

Pour remettre le lecteur en situation, nous allons tout d'abord rapidement rappeler les principaux objectifs et étapes qui constituent PANORAMA et permettent d'élaborer le Scénario de Mission Détaillé (SMD). Nous nous attarderons ensuite plus longuement sur les spécificités de l'axe de performance Localisation qui se démarque clairement des autres axes de performance déjà pris en compte. Le problème de localisation et d'affectation de ressources sera ensuite présenté. Après avoir formulé le problème nous montrerons que celui-ci peut être résolu en faisant appel à l'algorithme de Viterbi qui nous permettra de nous inscrire dans une logique de maximisation de la marge énergétique. Pour terminer, avant de conclure, l'approche finale SLED (*Sécurité-Localisation-Énergie-Durée*) mise en œuvre dans PANORAMA sera explicitée et comparée à l'approche SED (*Sécurité-Énergie-Durée*) actuelle sur un exemple didactique.

### 6.1 Retour vers PANORAMA(SED)

PANORAMA, détaillée dans le chapitre 2, est une méthodologie d'affectation des ressources matérielles et logicielles permettant de construire le scénario de missions robotiques autonomes garantissant la satisfaction des objectifs de performance imposés.

Elle nécessite de connaître au préalable la carte de l'environnement ainsi que le chemin nominal que le robot doit essayer de suivre. Par ailleurs, pour les axes correspondant à des performances pouvant évoluer dans le temps, on doit disposer de modèles prédictifs permettant de les projeter dans le futur. Les principales étapes méthodologiques sont alors les suivantes (Figure 6.1) pour une garantie SED (Sécurité - Énergie - Durée) :

1. Les différentes classes de contraintes connues (environnementales, méthodologiques, expérimentales, spatiales liées aux différents axes de performance, etc.) sont dans un premier temps projetées sur la mission pour définir des zones spatiales à contraintes constantes (Activités), limitant l'espace d'évaluation de certains paramètres, et ne préservant que les ressources restant à priori éligibles. Il reste alors à procéder à l'affectation des ressources.
2. A l'issue de cette étape l'axe Sécurité est satisfait, par construction, pour toutes les ressources présentes et leur paramétrage.
3. Après avoir trié en durée (loi prédictive pour un chemin donné à la vitesse maximale sécuritaire envisageable), pour chaque activité, toutes les alternatives d'implémentation envisageables, l'utilisation de l'algorithme de Bennour [30], permet de ne retenir qu'un sous-ensemble de ressources et de paramétrages permettant de satisfaire, à coup sûr, l'objectif de Durée imposé en fin de mission par l'utilisateur.
4. Pour terminer, après avoir réordonné en énergie (loi prédictive), pour chaque activité, les alternatives d'implémentation encore valides, l'algorithme de Bennour permet à nouveau de trouver quelle alternative doit être choisie au niveau de chaque activité pour satisfaire, en fin de mission la contrainte énergétique imposée.
5. On remarquera globalement que l'ordre de traitement des axes de performance n'est pas anodin. En effet, l'axe de performance Sécurité lié à un objectif cœur de mission, devant être satisfait tout au long de la mission est traité prioritairement, avant de prendre en compte les axes de performance Durée et Énergie liés à des objectifs fin de mission. Le choix inverse conduirait à statuer globalement sur des performances de fin de mission, avant d'avoir statué sur les contraintes locales à imposer au niveau des différentes activités, ce qui pourrait conduire à des erreurs d'allocation de ressources.

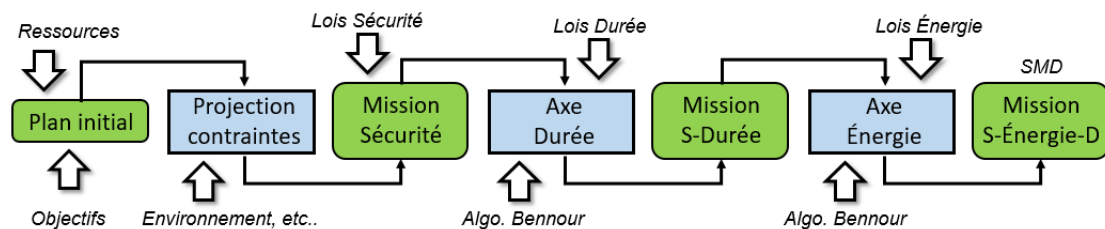


FIGURE 6.1 – PANORAMA(SED) : Principales étapes

L'utilisation de l'algorithme de Bennour pour résoudre ce problème d'allocation de type sac-à-dos suppose que toutes les activités sont indépendantes et que les lois de composition de performance entre activités préservent la relation d'ordre locale. Comme nous allons le constater dans la suite cette indépendance inter-activités ne pourra pas être préservée pour prendre en compte l'axe Localisation.

## 6.2 La localisation : Un axe de performance particulier

L'axe de performance Localisation possède un ensemble de spécificités qui le démarque des autres axes de performance que nous considérons. Ce paragraphe va nous permettre de les énoncer.

En premier lieu le principe d'indépendance inter-activités qui sous-tend la mise en œuvre de l'algorithme de Bennour suppose que la sélection des ressources d'une activité  $A_n$  est indépendante de celle de l'activité  $A_{n-1}$ . Il n'existe donc pas de corrélation entre deux activités successives, même si, bien évidemment, l'algorithme va effectuer des choix locaux au niveau

activité, pour satisfaire un critère global de performance au niveau mission. Ce principe d'indépendance inter-activités n'est pas vérifié pour l'axe de performance Localisation. En effet, pour qu'une méthode de localisation puisse être utilisée durant une activité  $A_n$ , il faut qu'elle ait été rendue préalablement opérationnelle si elle n'était pas déjà employée. Il faut en réalité un certain délai pour qu'une méthode de localisation soit opérationnelle et fournisse une localisation satisfaisant les contraintes imposées. Celui-ci dépend entre autres du temps qu'il faut pour les capteurs employés pour générer des données viables (images Kinect par exemple), du temps nécessaire à l'algorithme associé à la méthode visée pour délivrer une valeur de localisation pour le robot et du délai au bout duquel on va l'utiliser après qu'elle soit opérationnelle. Il existe donc, pour l'axe Localisation, une corrélation inter-activités.

Mais ce délai de préparation inter-activités induit un effet de bord impactant les performances finales en fin de mission. Dans le cas de l'axe de performance Énergie, il peut induire au niveau des activités précédant  $A_n$ , la mobilisation d'une méthode de localisation et des capteurs associés n'ayant aucune utilité, et n'ayant comme vocation que de préparer son utilisation pour l'activité  $A_n$ . Ce délai génère donc potentiellement (si la méthode de localisation considérée n'est pas déjà utilisée), au niveau des activités précédant  $A_n$ , un surcout énergétique tant en énergie liée au matériel, qu'au logiciel qui n'est pas pris en compte pour l'instant dans PANORAMA. Cette surconsommation imprévue peut, sur des missions de longue durée, fausser significativement la prévision de consommation énergétique, et violer notre objectif de garantie.

La méthodologie PANORAMA, dans sa configuration actuelle, sélectionne, pour l'axe Énergie, la combinaison d'alternatives d'implémentation la plus énergivore permettant de satisfaire globalement l'objectif de consommation énergétique imposé. Ce choix, pour le moins surprenant et contre intuitif, est dicté par le fait que le choix de la solution minorant la quantité d'énergie consommée aurait inévitablement conduit à sélectionner, tout au long de la mission, une localisation odométrique à l'estime. En effet ce dernier ne coûte rien énergétiquement puisque les données odométriques sont disponibles en permanence au niveau de la boucle de contrôle. Or, évidemment, la dérive odométrique rend ce choix inadapté si l'on veut préserver une bonne qualité de localisation. L'heuristique utilisée jusqu'à présent dans PANORAMA, supposait qu'en sélectionnant localement les solutions les plus énergivores, tout en satisfaisant la contrainte globale de consommation énergétique, on choisirait les méthodes de localisation les plus performantes. Cependant ce n'est qu'un postulat et rien ne garantissait que les méthodes choisies permettraient effectivement de localiser le robot avec la précision souhaitée. Le travail que nous avons développé depuis le début de ce manuscrit nous a permis de proposer des lois prédictives d'évolution de la précision de localisation (pose). Nous disposons donc à présent des éléments de base nous permettant de connaître et de projeter les capacités de localisation, et donc par transitivité, la consommation énergétique associée. Nous pouvons donc espérer pouvoir proposer une solution d'allocation de ressources, minimisant la consommation d'énergie, et donc maximisant la marge énergétique correspondante tout en satisfaisant l'ensemble des axes de performance, y compris celui de la Localisation.

Enfin, il est très important pour la méthodologie PANORAMA, d'être en mesure de suivre l'évolution réelle des marges performance par rapport à celles prédites. Cela permet d'accepter une certaine part de variabilité dans l'exécution de la mission, puisque, entre autre, le chemin à suivre ne sera jamais parfaitement suivi et que nos modèles sont par essence imparfaits. L'apparition d'une marge négative conduit actuellement, si cela est possible, à une reconfiguration de l'affectation des ressources jusqu'à la fin de la mission, où à l'échec, par anticipation, de cette dernière, s'il n'est plus possible de garantir son succès avec les contraintes imposées. Pour être à même de monitorer l'évolution réelle des marges de performance il faut pouvoir disposer

d'un mécanisme externe capable de mesurer, ou de déduire, avec une certaine précision, l'évolution de l'indicateur de performance analysé, depuis le début de la mission jusqu'à l'instant considéré. Pour l'axe Sécurité il n'existe pas de notion de marge de performance puisque la sécurité du robot est assurée par défaut. Pour l'axe Durée il est très facile d'utiliser l'horloge de l'ordinateur embarqué pour connaître le temps écoulé depuis le début de la mission avec une excellente précision. Pour l'axe Énergie nous disposons au niveau du robot d'un wattmètre embarqué mesurant la puissance instantanée consommée. En intégrant cette mesure dans le temps il est donc aisé de connaître l'énergie batterie consommée depuis le commencement de la mission. En ce qui concerne la batterie du PC embarqué, des fonctions Linux permettent aussi de suivre la consommation énergétique. En revanche, en ce qui concerne l'axe de Localisation, et en environnement intérieur, en l'absence de GPS, nous ne disposons pas d'un système externe omniscient, capable de localiser le robot avec une précision connue. Il faudra nous référer, lorsque cela est possible, à des amers externes de localisation absolue connue, pour avoir accès à une réalité terrain de précision limitée. Cela rend la garantie de Localisation extrêmement difficile à vérifier pratiquement tout au long d'une mission.

Nous venons de démontrer que l'axe de Localisation est un axe de performance très particulier. Le prochain paragraphe va nous permettre de formaliser le problème de l'allocation de ressources dans le cadre de la garantie de localisation. Nous y présenterons un algorithme permettant de résoudre efficacement le problème proposé tout en adressant la problématique de minimisation de consommation énergétique.

## 6.3 Localisation et Algorithme de Viterbi

Pour commencer précisons ce que désigne le terme "*Garantie de Localisation*". Cela ne veut absolument pas dire que nous cherchons à garantir que le robot suive le chemin nominal qui a été défini pour la mission. Cela n'aurait absolument aucun sens puisqu'il est clair qu'il ne sera jamais parfaitement en mesure de le faire, en raison d'une part, de l'inévitable différence entre la prescription et l'exécution et, d'autre part, en raison d'événements non planifiés (obstacles) conduisant le robot à modifier localement son chemin pour éviter un choc. La "*Garantie de Localisation*" désigne le fait que nous voulons être à même, durant la mission autonome, de garantir que nous connaissons la position (pose) du robot avec une précision/incertitude connue et définie à l'avance par l'utilisateur.

Dans un premier temps nous allons détailler le problème de Garantie de Localisation et son lien avec les lois prédictives qui ont été définies dans le chapitre précédent. Dans un second temps nous présenterons l'algorithme qui nous permettra de sélectionner localement, tout au long de la mission, pour chaque activité, les méthodes de localisation, qui permettront de satisfaire les contraintes imposées par l'utilisateur tout en adressant la problématique de maximisation de la marge énergétique.

### 6.3.1 Formalisation du problème

Tout comme les autres axes de performance, l'axe de Localisation nécessite de projeter les contraintes qui lui sont liées, sur le scénario initial de la mission. Celles-ci peuvent s'exprimer au travers des éléments suivants :

- Tout d'abord, bien évidemment, l'utilisateur doit définir spatialement, tout au long du scénario nominal de la mission, la garantie de Localisation qu'il souhaite voir satisfaite. Il est clair que celle-ci évolue durant l'exécution de la mission en fonction de tâches robotiques devant être exécutées. Dans certaines zones de transit on peut se satisfaire d'une

précision plus faible que dans les parties d'une mission nécessitant un positionnement de moindre incertitude lié à des actions plus ciblées. Ainsi, si l'on se réfère au vocabulaire que nous avons défini dans le chapitre 2 pour caractériser les axes de performance, la localisation est un axe de performance **principal**, puisqu'indispensable à la réalisation de toute mission robotique autonome. Il est associé à des objectifs **cœurs de mission variables** qui pourront donc évoluer au gré de son exécution en fonction des tâches robotiques mises en œuvre et de la précision de localisation correspondante exigée.

- Nous disposons d'un ensemble de Méthodes de Localisation  $M_i$  pré-caractérisées dans l'environnement de la mission. Lorsqu'elles sont éligibles nous sommes donc capable de connaître l'évolution du pavé d'incertitude de pose en fonction du déplacement attendu du robot.
- Par ailleurs nous supposons que nous sommes à même, dans cet environnement connu, et pour l'ensemble de méthodes de localisation  $M_i$  pré-caractérisées, et des capteurs associés, de définir spatialement, sur le scénario de mission initial, les zones où celles-ci peuvent être, ou non, employées.
- Enfin, l'utilisation de certaines méthodes peut aussi être subordonnée à la satisfaction de certaines contraintes. Par exemples l'utilisation d'un capteur de type caméra peut nécessiter un niveau d'éclairage minimal, ou imposer une limitation de la vitesse du robot de façon à réduire les vibrations mécaniques et ainsi préserver la qualité du flux vidéo et permettre son traitement.

Avant d'aborder la formalisation du problème attardons nous un peu sur les délais mis en jeu séquentiellement pour utiliser une méthode de localisation donnée. Ceux-ci peuvent s'expliquer au travers de la Figure 6.2 où en bleu on trouve des instants de recalage potentiels.

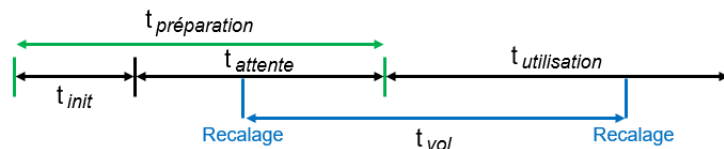


FIGURE 6.2 – Utilisation d'une méthode de localisation : Délais mis en jeu

On distingue les durées suivantes :

- Le temps d'initialisation  $t_{init}$  désigne la durée nécessaire pour obtenir des informations de localisation viables. Cela dépend donc du temps d'opérationnalisation des capteurs impliqués dans une méthode s'ils ne sont pas déjà utilisés et des algorithmes liés à la mise en œuvre de celle-ci.
- Le temps d'attente  $t_{attente}$  correspond à la durée séparant la fin du temps d'initialisation et le début d'utilisation effective d'une méthode. Une fois la méthode initialisée son profil d'erreur va donc suivre les lois prédictives établies.
- Le temps de préparation  $t_{preparation}$  correspond donc à la somme de  $t_{init}$  et  $t_{attente}$ . Sa durée est toujours établie en prenant en compte le dernier recalage possible avant l'utilisation d'une méthode.
- Le temps d'utilisation  $t_{utilisation}$  correspond à la durée pendant laquelle la méthode restera employée. Pendant cette période le profil d'évolution de l'incertitude de localisation est donc compatible avec les contraintes de pose fixées.
- Si la méthode fait appel à un recalage extéroceptif, le temps de vol  $t_{vol}$  correspond au temps entre 2 recalages successifs. Cette durée peut être variable mais il est possible de connaître sa valeur maximale.



Compte tenu de ces notations on peut distinguer 2 cas de figure en fonction de la classe de méthodes de localisation utilisées.

- Soit on fait appel purement à des méthodes de navigation à l'estime (odométrie, inertielle). Celles-ci sont en général préalablement initialisées une fois pour toutes avant le début de la mission ce qui induit un temps d'attente nul. Elles restent donc toujours directement utilisables durant la mission, sans temps de préparation, et fournissent des informations de localisation régulièrement à une fréquence donnée.
- Soit on fait appel à des méthodes de suivi de posture ou de localisation globale. Les informations de recalage ponctuelles vont alors être combinées avec les informations de suivi de mouvement intéroceptives entre 2 recalage successifs. Dans ce cas de figure il faut donc au préalable initialiser matériellement et logiciellement la méthode et attendre le moment où elle sera effectivement utilisée.

Dans ce dernier cas, cette latence entre l'instant où on lance une méthode de localisation et l'instant où on l'utilise effectivement aura, comme le traduit la Figure 6.3, un ensemble d'incidences qu'il nous faudra prendre en compte :

- Tout d'abord pour qu'une méthode ML puisse être utilisée au niveau d'une activité il faut qu'elle soit éligible en permanence durant toute la durée  $t_{préparation}$  de sa préparation. On ne peut donc pas lancer une méthode de localisation de type suivi de posture ou localisation absolue ex-nihilo, il faut qu'elle soit au moins éligible dans l'activité précédent son utilisation.
- Évidemment la préparation d'une méthode de localisation durant  $t_{préparation}$  induit une surconsommation énergétique. Celle-ci sera évaluée et prise en compte globalement en une seule fois au niveau de la dernière activité précédant l'utilisation de la méthode.
- De la même façon cette préparation induit d'une part, au niveau du scénario final SMD, une modification des Alternatives d'implémentation MA(ML) (Figure 6.3) pour les activités précédant l'utilisation de la méthode. D'autre part elle va conduire à la création d'une nouvelle activité (A12) si l'instant de début d'initialisation ne correspond pas exactement à un début d'activité du scénario initial.

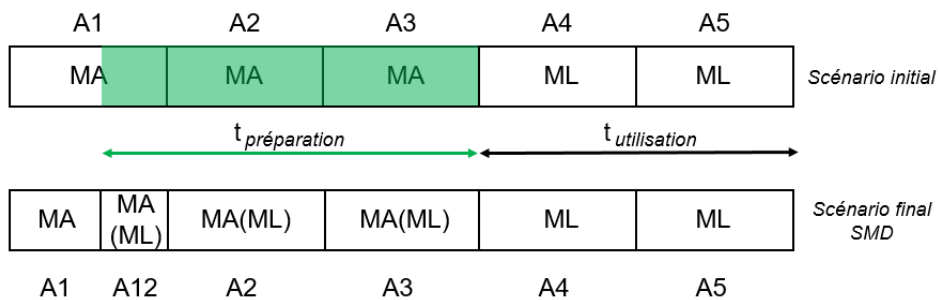


FIGURE 6.3 – Impact du temps de préparation d'une méthode de localisation

Dès lors si l'on restreint notre analyse au seul axe de Localisation et aux contraintes que nous venons d'énoncer, la décomposition du scénario de la mission en activités de contraintes constantes peut s'illustrer au travers de la Figure 6.4.

Cette figure représente le scénario d'une mission selon son abscisse  $X_L$  en coordonnées linéaires 2D. On trouve tout en haut, selon cette abscisse, les domaines spatiaux d'éligibilité des méthodes de localisation (M1 à M5). On distingue donc 3 zones distinctes. La ligne continue rouge en bas de la figure représente en fonction de  $X_L$  l'évolution de l'incertitude de localisation  $P_{L_i,max}$  acceptée par l'utilisateur. De fait cette courbe est évidemment tri-dimensionnelle puisque l'utilisateur doit définir une incertitude en X, Y et  $\theta$ . On constate que l'on définit 3

zones, celle centrale correspondant à une demande d'accroissement de la précision de localisation souhaitée. L'intersection entre les contraintes d'éligibilité des méthodes, et les contraintes de localisation conduit à définir 5 activités ( $A_i$  à  $A_{i+4}$ ) correspondant donc à des contraintes constantes. Pour chacune d'elles, le problème de Garantie de Localisation qui nous intéresse doit conduire à choisir parmi les méthodes éligibles d'une activité celle ( $M_i$  à  $M_{i+4}$ ) qui permettra de satisfaire la contrainte d'incertitude fixée. La courbe verte du bas représente un exemple théorique d'évolution de la précision de Localisation pour une sélection de méthodes.

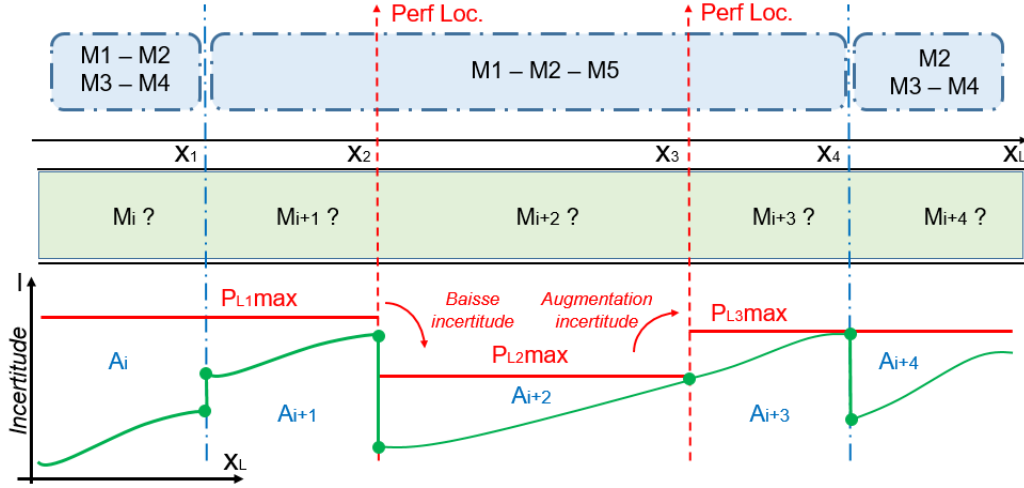


FIGURE 6.4 – Scénario de mission et performance de Localisation

Le problème peut donc s'énoncer de la façon suivante "Quelle méthode de Localisation  $M_i$  doit être sélectionnée au niveau de chaque activité du scénario de mission de façon à satisfaire les objectifs locaux de localisation  $P_{L_i,max}$  tout au long de la mission envisagée?"

Soit une activité  $A_i$ , commençant et s'achevant respectivement à une abscisse linéaire D et F. Si l'on sélectionne une méthode de localisation  $M_i$  pour cette activité alors elle doit permettre de maintenir l'incertitude de pose  $I$  du robot à une valeur inférieure à  $P_{L_i,max}$  durant toute l'activité (Figure 6.4). Cela se traduit alors par l'ensemble d'équations suivantes (Équation 6.1) :

$$\begin{cases} (I(M_i, X, x_L)) + K_{M_i} \cdot I_{max_X}(A_{i-1}) \leq I_{max_X}(A_i) \\ (I(M_i, Y, x_L)) + K_{M_i} \cdot I_{max_Y}(A_{i-1}) \leq I_{max_Y}(A_i) \\ (I(M_i, \theta, x_L)) + K_{M_i} \cdot I_{max_\theta}(A_{i-1}) \leq I_{max_\theta}(A_i) \end{cases} \quad \forall D_{A_i} \leq x_L \leq F_{A_i} \quad (6.1)$$

Dans ce système d'équations on utilise les conventions d'écritures suivantes :

- $(X, Y, \theta)$  désigne la pose du robot dans le référentiel Monde  $\mathcal{R}_M$ .
- $I(M_i, C_M, x_L)$  représente l'incertitude selon la composante  $C_{M_i}$  de la pose du robot dans  $\mathcal{R}_M$  associée à l'utilisation de la méthode de localisation M au sein de l'activité  $A_i$  au point de coordonnée linéaire  $x_L$ . Le robot suivant une trajectoire curviligne dont la projection correspond à une valeur  $x_L$  en coordonnées linéaires 2D, on vérifie que  $D_{A_i} \leq x_L \leq F_{A_i}$ .
- $K_{M_i}$  désigne un coefficient booléen dépendant de la nature de la méthode de localisation dans l'activité  $A_i$ .
- $I_{max_{C_M}}(A_i)$  désigne l'incertitude maximale admise par l'utilisateur selon une composante  $C_M$  de la pose du robot dans  $\mathcal{R}_M$  pour l'activité  $A_i$ .

Pratiquement cela conduit donc à 2 formulations en fonction de la nature des méthodes de localisation considérées :

- Soit c'est uniquement une méthode de localisation à l'estime alors dans ce cas de figure  $K_{M_i} = 1$ . On ajoute donc l'incertitude maximale admissible au sein d'une activité en fin de l'activité précédente à l'incertitude de mouvement. En effet la vérification de la satisfaction des contraintes de localisation se fait indépendamment de la performance de localisation obtenue dans l'activité précédente. Cela correspond donc nécessairement à un accroissement d'incertitude par rapport à l'activité précédente. Le profil d'incertitude sera donc du type de celui observé au niveau des activités  $A_{i+3}$  de la Figure 6.4.
- Soit c'est une méthode de suivi de posture ou de localisation absolue nécessitant donc une durée de préparation. D'une part l'évolution de l'incertitude de localisation de la méthode  $M_i$  est indépendante de celle de l'activité précédente et donc  $K_{M_i} = 0$ . D'autre part on vérifie pour  $M_i$ , en prenant en compte le délais de préparation et le dernier recalage possible avant son utilisation, que nos modèles prédictifs permettent d'assurer que  $(I(M_i, C_M, x_L)) \leq I_{max_{C_M}(A_i)} \forall D_{A_i} \leq x_L \leq F_{A_i}$ . Il est donc possible d'observer une décroissance de l'incertitude de localisation comme c'est le cas pour le profil d'incertitude dans la Figure 6.4 pour les activités  $A_{i+2}$  et  $A_{i+4}$  ou à l'inverse un accroissement comme pour  $A_{i+1}$

Le cadre du problème de sélection d'une méthode de localisation étant fixé nous allons maintenant présenter l'algorithme qui va nous permettre de le résoudre.

### 6.3.2 Algorithme de Viterbi

Nous sommes à la recherche d'un algorithme qui viendrait compléter la démarche PANORAMA actuelle en permettant d'intégrer explicitement la prise en compte de la performance de Localisation tout en permettant de s'inscrire dans une logique de maximisation de la marge énergétique. Notre regard s'est tourné vers l'algorithme proposé dans un tout autre contexte par A. Viterbi en 1967.

L'algorithme de Viterbi [224] [83], est connu en programmation dynamique, pour sa capacité à identifier le "chemin de Viterbi" lorsqu'un problème peut être modélisé sous la forme d'un treillis. Comme le filtre de Kalman, cet algorithme peut traquer récursivement les états d'un processus stochastique. Il ne s'intéresse toutefois pas aux lois Gaussiennes mais a été initialement développé pour les modèles de Markov cachés dans lesquels certains états ne sont pas observables (*Hidden Markov Model*) mais où certaines informations peuvent être déduites à partir des données observées. Le chemin de Viterbi correspond alors à la séquence la plus probable d'états cachés conforme aux observations. Ce problème est isomorphe à celui, plus classique, de trouver le chemin à coût minimal au sein d'un treillis. Cet algorithme a trouvé de nombreuses applications au sein de champs disciplinaires variés tels que la linguistique et le traitement automatisé du langage, la bio-informatique, le machine-learning, la robotique avec la localisation [148] ou la capture de déplacements humains [160].

Si l'on considère un problème pouvant être modélisé par le treillis de la Figure 6.5 composé de  $N$  états (largeur du treillis) et par une séquence de longueur  $M$  (longueur du treillis) alors il existe  $N^M$  séquences d'états envisageables. L'approche de programmation dynamique proposée par Viterbi permet de calculer le chemin de Viterbi avec une complexité linéaire en temps en  $O(N^2M)$ . Sous certaines hypothèses cette complexité peut encore être améliorée [22].

Afin de présenter plus en détail l'algorithme de Viterbi, considérons la formulation de la recherche du chemin de Viterbi sur la Figure 6.5. Le treillis peut-être décomposé en  $M$  sections

de  $N$  nœuds (états). Chaque arête correspond aux liens existants entre les nœuds de 2 sections voisines. Il est possible de distinguer 2 classes de coût dans ce treillis. D'une part le paramètre  $C_s(i, j)$  est associé à un coût incrémental correspondant, pour la section  $s$ , à la transition menant du nœud  $i$  au nœud  $j$ . D'autre part  $W_s(i, a)$  représente quant-à lui le coût associé au nœud  $i$  pour la section  $s$ .

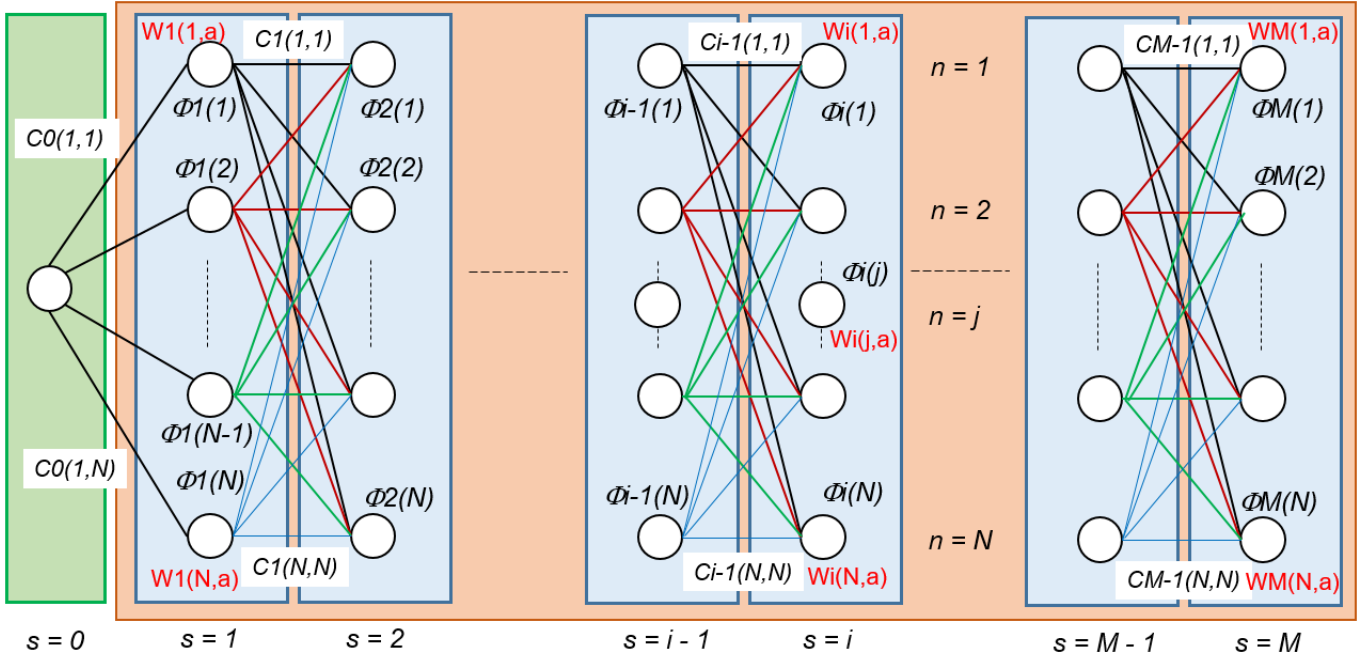


FIGURE 6.5 – Treillis et algorithme de Viterbi

Si l'on se projette dans le cas de la représentation d'un modèle Markovien caché alors le coefficient  $C(i, j)$  représente la probabilité de transition permettant d'aller d'un état  $i$  à un état  $j$ .  $W(i, a)$  correspond alors à la probabilité de voir un symbole  $a$  appartenant à un alphabet  $[\sigma]$  de dimension  $\sigma$  dans l'état  $i$ , soit  $S = (a_1, \dots, a_M)$  une séquence de  $M$  observations où  $a_t \in [\sigma]$ , soit la séquence d'états  $U = (U_0, U_1, \dots, U_M)$  débutant avec  $U_0 = 1$ , alors l'algorithme de Viterbi détermine la séquence  $U$  maximisant la probabilité conditionnelle  $P(U|S)$ .

Ce problème peut être modélisé comme un problème d'optimisation formulé par le biais de l'Équation 6.2 suivante :

$$U = \underset{U_0, U_1, \dots, U_M}{\text{Arg max}} \prod_{s=1}^M [C(U_{s-1}, U_s) \cdot W(U_s, a_s)] \quad (6.2)$$

L'application d'une loi logarithmique à cette formulation permet de transformer le problème de maximisation d'une probabilité en un problème de minimisation selon l'Équation 6.3.

$$U = \underset{U_0, U_1, \dots, U_M}{\text{Arg min}} \sum_{s=1}^M [C(U_{s-1}, U_s) + W(U_s, a_s)] \quad (6.3)$$

Si  $W(U_s, a_s) = 0$  alors l'algorithme de Viterbi permet de résoudre le problème du plus court chemin entre 2 points. Chaque chemin de  $s = 1$  à  $s = M$  a un coût associé qui est égal à la somme des coûts de tous les nœuds et arêtes qu'il traverse. L'algorithme de Viterbi permet de trouver le chemin minimisant ce coût.

Détaillons maintenant plus précisément la mécanique de l'algorithme de Viterbi en présentant son pseudo-code. Il peut se décomposer en 3 parties distinctes, l'initialisation, l'induction

qui parcourt récursivement le treillis, et enfin, une fois celui-ci traversé, le rassemblement qui construit le meilleur chemin le parcourant.

Tous les coefficients de coût  $C(i, j)$  et  $W(i, j)$  du treillis ainsi que la structure de ce dernier étant supposés connus, l'étape d'initialisation (Algorithme 6) permet d'initialiser, au niveau de chacun des états de chaque section, une variable *Memo* qui sera utilisée pour mémoriser le coût du plus court chemin menant à un état ainsi que l'état précédent permettant d'obtenir ce minimum.

---

**Algorithme 6 :** Algorithme de Viterbi : Étape d'initialisation

---

```

1 for  $i = 0$  to  $M$  (pour toutes les sections) do
2   | for  $j = 1$  to  $N$  (pour tous les états) do
3   |   |  $Memo(i, j) = (0, -1)$  (coût initial = 0, aucun état précédent sélectionné)
4   |   end
5 end

```

---

L'étape d'induction (Algorithme 7) constitue le cœur même de l'algorithme de Viterbi et de la programmation dynamique dont le principe est de décomposer un problème en sous-problèmes, tout en stockant des résultats intermédiaires qui permettront, in fine, lors de la phase de rassemblement, de résoudre globalement le problème général posé. Dans notre cas, pour chaque état de chaque section on commence par identifier le chemin à coût minimal conduisant à un état courant  $j$  d'une section  $i$ . Pour cela on utilise une fonction de coût  $\Phi_i(j)$  qui ajoute au coût minimal  $\Phi_{i-1}(n)$  de tous les chemins menant aux états de la section précédente liés à  $j$ , le coût de transition d'un de ces états ( $C_{i-1}(n, j)$ ) à l'état  $j$ . Une fois le coût minimal de tous les chemins menant à  $j$  déterminé, la fonction  $\Phi$  ajoute le coût de l'état considéré ( $W_i(j, a)$ ). Au niveau d'un état considéré d'une section  $i$ , on ne garde alors en mémoire que le coût du chemin minimal ainsi que l'état de la section précédente permettant de l'obtenir ( $n_{j,i-1}$ ). La progression de l'algorithme se fait de la gauche vers la droite du treillis c'est à dire en largeur d'abord, jusqu'à ce que tous les états du bord droit aient été traités.

---

**Algorithme 7 :** Algorithme de Viterbi : Étape d'induction

---

```

1 for  $i = 0$  to  $M$  (pour toutes les sections) do
2   | for  $j = 1$  to  $N$  (pour tous les états) do
3   |   |  $\Phi_i(j) = \min_{1 \leq n \leq N} [\Phi_{i-1}(n) + C_{i-1}(n, j)] + W_i(j, a)$  (Calcul de la fonction de coût)
4   |   |  $Memo(i, j) = (\Phi_i(j), n_{j,i-1})$  (Mémorisation du coût minimal et de l'état précédent)
5   |   end
6 end

```

---

L'étape de rassemblement (Algorithme 8) va reconstruire le chemin minimal, de proche en proche, de la droite vers la gauche du treillis, à partir des informations mémorisées au niveau des états.

Au niveau de la dernière section du treillis (section  $M$ ), on identifie tout d'abord l'état conduisant au coût minimal  $\Phi_M(j_{min})$ . Il est alors possible grâce au couple mémorisé correspondant à savoir  $Memo(M, j_{min}) = (\Phi_M(j_{min}), n_{j_{min}, M-1})$ , de remonter à l'état le précédent. On procède alors itérativement jusqu'à la première section. L'Algorithme 8 est utilisé pour reconstruire le chemin minimal. Pour le mémoriser, on complète alors 3 vecteurs de  $M$  éléments. Un vecteur  $U$  qui mémorise les états que le chemin minimal traverse. Un vecteur  $W$  qui

conserve le coût associé à chacun des états traversés. Enfin, un vecteur  $C$  qui contient les coûts incrémentaux entre 2 états successifs du chemin minimal.

---

**Algorithme 8 :** Algorithme de Viterbi : Étape de rassemblement

---

```

1  $U[M] = j_{min}$ 
2  $W[M] = W_M(j_{min}, a)$ 
3  $C[M] = C_{M-1}(n_{j_{min}, M-1}, j_{min})$ 
4  $Current = n_{j_{min}, M-1}$ 
5 for  $i = M - 1$  to 1 (pour toutes les sections restantes) do
6    $CurrentMemo = Memo(i, Current)$  (Mémorisation des informations liées à l'état courant du chemin minimal)
7    $Previous = CurrentMemo[1]$  (Récupération l'état précédent du chemin minimal)
8    $U[i] = Current$  (Mémorisation, dans le vecteur U, de l'état courant du chemin minimal)
9    $W[i] = W_i(Current, a)$  (Mémorisation, dans le vecteur W, du coût de l'état courant du chemin minimal)
10   $C[i] = C_i(Previous, Current)$  (Mémorisation, dans le vecteur C, du coût de passage entre Current et Previous)
11   $Previous = Current$  (On remonte vers la gauche)
12 end

```

---

L'algorithme de Viterbi ayant été détaillé dans sa forme générale, nous allons maintenant présenter comment il peut être utilisé, lors d'une mission robotique pour sélectionner les méthodes de localisation, et ainsi les alternatives d'implémentation, devant être utilisées tout en permettant de maximiser la marge énergétique.

### 6.3.3 Localisation, marge énergétique et algorithme de Viterbi

Nous sommes donc à la recherche d'un algorithme pouvant s'adapter facilement à la formalisation du problème de garantie de localisation tout en étant capable de proposer une solution minimisant le coût en énergie d'une mission et donc, par conséquence, maximisant la marge énergétique correspondante.

#### 6.3.3.1 Garantie de Localisation

L'algorithme de Viterbi pourrait être adapté à l'ensemble des axes de performance mais nous nous focalisons ici sur l'axe de Localisation. Dans ce contexte, un ensemble de données permet de déterminer, par activité, les méthodes de localisation éligibles. On dispose pour cela des informations initiales suivantes :

- On connaît spatialement, sur le scénario prévisionnel de la mission, les contraintes d'incertitudes de pose imposées par l'utilisateur. Celles-ci dépendent bien évidemment des actions que le robot doit réaliser.
- L'analyse qui a été réalisée pour chacune des méthodes de localisation permet de connaître, sur la carte de l'environnement, les zones où elles peuvent être (ou non) employées. Cela dépend à la fois des caractéristiques de l'environnement, des capteurs sollicités, et de la nature même de la méthode considérée.

- Grâce aux travaux précédents, nous disposons, pour chacune des méthodes disponibles, de modèles prédictifs d'évolution de la performance de localisation et cela pour les 3 dimensions de la pose.

A présent, à partir de ces informations, l'objectif est double. D'une part déterminer spatialement, sur le scénario de la mission, les activités induites par la sélection des méthodes de localisation. D'autre part, ne retenir que celles qui permettront de vérifier les contraintes de localisation imposées. Pour ce faire on va procéder de la façon suivante :

- Pour déterminer les activités nous allons procéder comme nous le faisons classiquement dans PANORAMA pour les autres axes de performance. On projette sur le scénario prévisionnel de la mission les zones d'éligibilité de chacune des méthodes de localisation disponibles. Une activité, au sens de la localisation est alors définie comme une zone à l'intérieur de laquelle l'ensemble des méthodes éligibles reste le même pour des contraintes de localisation invariables.
- Connaissant pour chacune des activités les contraintes de localisation et les modèles prédictifs des méthodes éligibles il est alors simple de vérifier si le choix d'une méthode de localisation, permet de satisfaire les contraintes de pose imposées en fin d'activité. Si c'est le cas la méthode considérée peut être employée. Dans le cas contraire elle est supprimée de la liste des méthodes éligibles pour l'activité considérée.

A l'issue de ce processus nous disposons donc, pour chacune des activités du scénario de mission d'une liste de méthodes de localisation non plus éligibles mais utilisables. Il nous faut maintenant sélectionner, tout au long de la mission celles qui nous permettront de minimiser le coût énergétique. C'est ce qui va nous amener à chercher à résoudre un problème dit "du plus court chemin" entre 2 points.

### 6.3.3.2 Maximisation de la marge énergétique

De nombreux algorithmes ont été proposés pour résoudre ce type de problème. L'algorithme de Dijkstra [69] permet d'y répondre lorsque le graphe est composé d'arcs à poids positifs. L'algorithme  $A^*$  [100] en est une variante très connue qui utilise une heuristique pour estimer le meilleur chemin passant par chaque nœud. Celui de Bellman-Ford [28] [82] permet de prendre en compte des arcs à poids négatifs.

Nous retiendrons celui de Viterbi que nous avons déjà présenté en détail. En effet, s'appliquant à des modèles en treillis il se trouve directement adapté à la problématique de sélection de scénario de mission à garantie de localisation qui nous intéresse. D'autre part, la programmation dynamique à laquelle il fait appel est une méthode incrémentale minimisant le coût mémoire, ce qui peut être avantageux pour des applications embarquées.

En s'appuyant sur la Figure 6.5 et ses notations les correspondances suivantes peuvent être effectuées entre la structure de treillis et le problème de garantie de localisation dans l'objectif de maximiser la marge énergétique.

- La longueur  $M$  du treillis correspond au nombre d'activités de la mission une fois les contraintes cœur de mission prises en compte. En conséquence, chaque section du treillis correspond à une activité du scénario de mission et donc à un espace où les contraintes sont constantes.
- $S = 0$  représente l'état de départ qui n'a de signification que pour le déploiement de l'algorithme. Cet état est important car on suppose qu'il permet de préparer toutes les méthodes éligibles au niveau de la section 1. Il doit être distingué des états initiaux de la section 1 qui constituent effectivement ceux associés à la première activité du scénario de la mission robotique. Toutes les méthodes de localisation des états initiaux sont

donc directement utilisables quelles que soient leur nature, navigation à l'estime, suivi de posture, localisation absolue. Compte tenu de leur préparation hors lignes, les poids énergétiques des arcs entre l'état de départ et les états initiaux sont considérés nuls. Par ailleurs on suppose évidemment que nous connaissons la pose initiale de notre robot avec une incertitude connue.

- Un état correspond à une Alternative d'Implémentation (Cf Chapitre 2) c'est à dire la sélection d'une unique combinaison de ressources logicielles et matérielles permettant d'exécuter les tâches robotiques devant être mise en œuvre pour l'activité considérée. Nous rappelons encore que nous ne ferons référence ici qu'à la sélection des seules méthodes de localisation.
- Conformément à ce que nous venons de rappeler, les  $N$  états d'une section correspondent aux  $N$  méthodes de localisation éligibles pour une activité donnée.
- Pour une activité (section), un état est caractérisé par l'énergie consommée par la méthode de localisation considérée, par la durée de cette activité (vitesse maximale envisageable) et les capteurs mobilisés bien évidemment. Cette énergie correspond au coût  $W_s(i, a)$  associé au nœud  $i$  d'une section  $s$ .
- De façon à maximiser la marge de performance de Durée, et donc minimiser la durée d'une activité, on sélectionne toujours, pour une alternative d'implémentation, la vitesse maximale compatible avec l'exécution des tâches robotiques.
- Les arcs reliant deux nœuds de sections voisines sont associés à un poids  $C_s(i, j)$  correspondant, pour la section  $s$ , au coût incrémental de transition entre le nœud  $i$  de  $s$  et le nœud  $j$  de  $s + 1$ . Ce coût représente le coût de préparation de la méthode de localisation ( $j$ ) qui peut être utilisée au niveau de la section  $s + 1$ . Il faut souligner que ce coût de préparation énergétique concentre toute l'énergie dissipée durant le temps de préparation qui temporellement peut débuter bien avant la section  $s$ .
- Dans notre problème, la structure du treillis considéré est bien moins complexe que celui présenté dans la Figure 6.5. En effet si un arc relie 2 états  $i$  et  $j$  faisant appel à la même méthode de localisation alors son coût de préparation est nul  $C_s(i, j) = 0$  puisqu'il n'existe pas de délais de préparation (6.6).

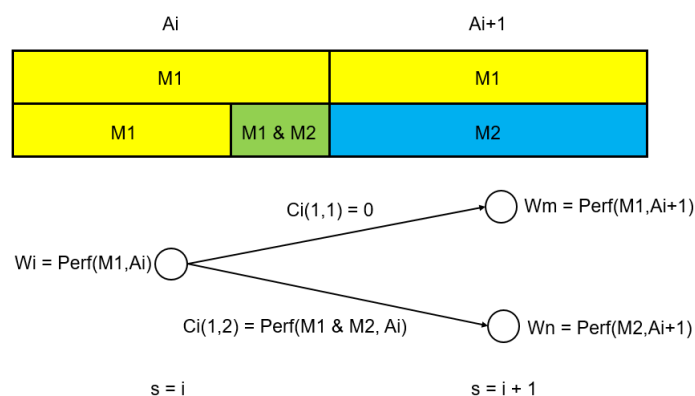


FIGURE 6.6 – Algorithme de Viterbi : Scénario de Mission et délais de préparation

La Figure 6.6, légèrement simplifiée car un état peut avoir plusieurs arcs d'entrée, illustre donc l'impact énergétique induit en présence d'une corrélation inter-activité.  $\text{Perf}(Mk, A_j)$  représente le coût énergétique lié au choix de la méthode de localisation  $Mk$  pour l'activité  $A_j$ . On constate que pour l'activité  $A_{i+1}$  on peut choisir soit  $M1$  soit  $M2$  en partant de l'état sélectionnant  $M1$  dans l'activité  $A_i$ . En conséquence dans le premier cas le coût de préparation de la méthode  $M1$  sera nul alors que dans le second cela induira l'ajout d'une surconsommation



$Perf(M1\&M2, A_i)$ . Si maintenant on s'intéresse au scénario de mission, on peut observer que l'introduction d'un délai de préparation pour la méthode  $M2$  induit la création, ex-nihilo, entre l'activité  $A_i$  et l'activité  $A_{i+1}$  d'une nouvelle activité où les 2 méthodes  $M1$  et  $M2$  devront être lancées.  $M1$  sera effectivement employée par le robot pour se localiser, alors que  $M2$  sera en phase d'opérationnalisation pour pouvoir se substituer à  $M1$  pour  $A_{i+1}$ . Il faudra considérer ces cas de figure pour la création du scénario final de la mission.

À la lecture de ce que nous venons d'énoncer, on constate donc la parfaite adéquation de la structure en treillis avec le problème de garantie de localisation posé. Il suffit maintenant de mettre en œuvre l'algorithme de Viterbi sur le modèle obtenu pour déterminer la séquence d'états  $(U_1, \dots, U_N)$  satisfaisant à la fois les contraintes Sécuritaire, de Durée de mission, de Localisation imposées tout en maximisant la marge énergétique. Évidemment, tout comme dans les travaux précédents de l'équipe EXPLORE, il faudra monitorer, tout au long de la mission, face aux événements imprévus et aux imprécisions de nos modèles prédictifs, l'évolution des marges de durée et énergétique, pour vérifier que la mission peut s'achever en maintenant la garantie imposée.

En maximisant la marge énergétique comme nous venons de le faire, contrairement aux travaux présentés dans [112] il sera alors impossible de trouver des solutions de repli en cas de violation. Une situation intermédiaire peut être imaginée puisque l'algorithme de Viterbi permet de choisir une fonction d'utilité quelconque. Au lieu de nous appuyer uniquement sur la dimension énergétique on peut imaginer à la fois la mixer et la pondérer avec la dimension Durée, tout en limitant les marges choisies (utilisées) par rapport aux marges maximales atteignables (utilisables). Il serait alors possible, en cas de violation des marges, de trouver des situations de repli, en faisant appel aux marges préservées encore utilisables. Il faut cependant rappeler qu'une remise en cause d'un scénario de mission peut toujours intervenir en cas de dysfonctionnement matériel et/ou logiciel rendant inopérant une des alternatives d'implémentation sélectionnée.

Nous venons de présenter comment la problématique de garantie de Localisation peut être prise en compte tout en maximisant la marge énergétique. Nous allons maintenant expliciter son intégration au sein de la méthodologie PANORAMA et analyser ses déclinaisons SED et SLED sur un exemple didactique.

## 6.4 PANORAMA(SLED) : Garantie de performance SLED

La mise en œuvre de l'algorithme de Viterbi va venir compléter la démarche PANORAMA de garantie SED proposée dans [112] et résumée par la Figure 6.1. La démarche PANORAMA(SLED) garantissant les contraintes de performance suivant les axes Sécurité - Localisation - Énergie - Durée peut être déclinée selon 2 alternatives en fonction de l'approche retenue pour initier la prise en compte des contraintes énergétiques. Nous allons maintenant les présenter.

### 6.4.1 Méthodologie finale SLED1

La méthodologie SLED1 peut être synthétisée par le diagramme de la Figure 6.7. Elle est constituée des principales étapes méthodologiques suivantes :

1. Tout comme pour PANORAMA(SED) les contraintes sur les axes correspondant à des objectifs cœur de mission sont d'abord considérées.

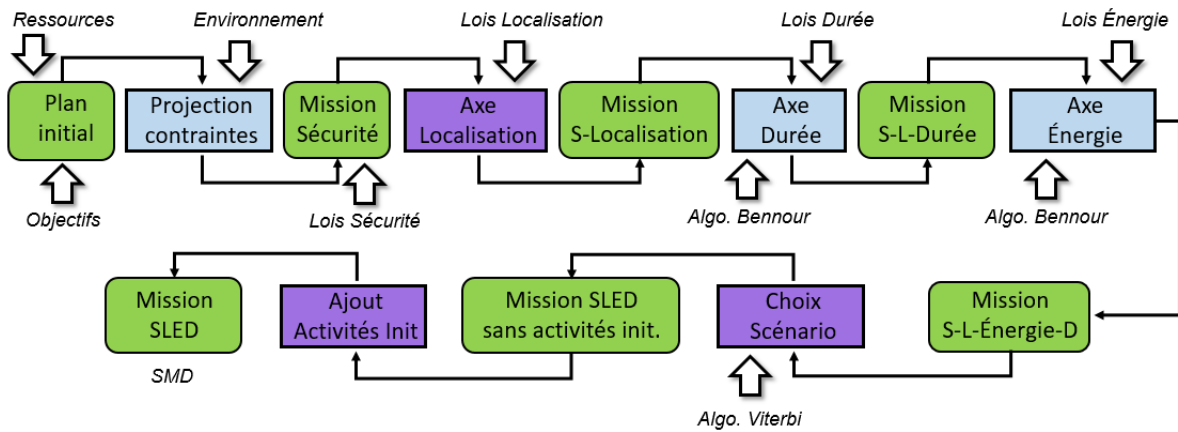


FIGURE 6.7 – PANORAMA(SLED1) : Principales étapes

2. Après avoir pris en compte les contraintes de Sécurité, on sélectionne donc, en faisant appel aux modèles prédictifs établis, les méthodes de Localisation à même d'assurer les contraintes de Localisation imposées. Pour chacune des méthodes de localisation éligibles et pour chacune des dimensions de la pose  $(x,y,\theta)$  on élimine celles ne permettant pas d'atteindre la performance de localisation visée en fin d'activité en supposant que l'on débute celle-ci avec la moins bonne performance issue de l'activité précédente. A l'issue de ce filtrage, chacune des méthodes de localisation encore présente correspond, pour une activité, à un coût énergétique incluant la phase de préparation.
3. Après avoir filtré les alternatives d'implémentation en Durée grâce à l'algorithme de Bennour, ce dernier est ensuite utilisé pour l'axe de performance Énergie, comme pour PANORAMA(SED). La différence est qu'au lieu de sélectionner, au niveau de chacune des activités, l'alternative d'implémentation la plus énergivore permettant de ne pas violer globalement la contrainte énergétique imposée, on utilise cet algorithme pour connaître, pour chaque activité, les sous-ensembles d'alternatives d'implémentation dont on est certain que la composition ne violera pas la contrainte énergétique globale à satisfaire.
4. S'il reste des alternatives éligibles, l'algorithme de Viterbi est ensuite utilisé pour sélectionner au niveau de chaque activité les alternatives d'implémentation qui devront effectivement être employées lors de l'exécution de la mission. La fonction d'utilité permet alors soit de limiter les marges de performances utilisées, soit de les maximiser. Dans le cadre de ce document nous nous limiterons à la maximisation des marges de Durée (par défaut) et Énergétique.
5. Une fois la sélection des alternatives d'implémentation obtenue grâce à l'algorithme de Viterbi, il faut encore redécouper le scénario de mission considéré depuis la projection des contraintes en introduisant, lorsque cela est nécessaire, les activités permettant la préparation des méthodes de Localisation, pour obtenir le Scénario de Mission Détaillé (SMD). Remarquons ici que tout le filtrage énergétique préalable a été effectué en considérant, pour chaque activité, qu'une méthode devait être préparée. Cela ne remet donc pas en cause le résultat de l'algorithme de Viterbi puisque ce dernier considère une consommation énergétique "pire cas" qui ne pourra que décroître lorsque le délai d'opérationnalisation d'une méthode de Localisation n'est pas indispensable.

Arrêtons nous maintenant sur la nature de la contrainte énergétique qui doit être imposée en entrée de la méthodologie SLED1. Si l'on analyse la démarche présentée, celle-ci intègre immédiatement et systématiquement, pour toute méthode de localisation, l'énergie d'initialisation qui lui est associée comme s'il fallait nécessairement la lancer au niveau de chaque activité. Cela revient à dire que, pour que des alternatives restent éligibles pour la mission S-L-Énergie-

Durée, on doit intégrer dans la limite énergétique initialement fixée, une surestimation du coût énergétique réel obtenu à l'issue de l'application de l'algorithme de Viterbi. Expérimentalement il s'avère que cette surestimation du coût énergétique d'entrée peut conduire à accroître de plusieurs ordres de grandeur la contrainte en énergie si l'on veut pouvoir obtenir une Mission SLED. Cette surestimation est problématique car alors la contrainte énergétique d'entrée devient totalement décorrélée de la réalité finale. Pour pallier cet effet de bord une seconde démarche SLED peut être proposée.

## 6.4.2 Méthodologie finale SLED2

La méthodologie SLED2 peut être synthétisée par le diagramme de la Figure 6.8.

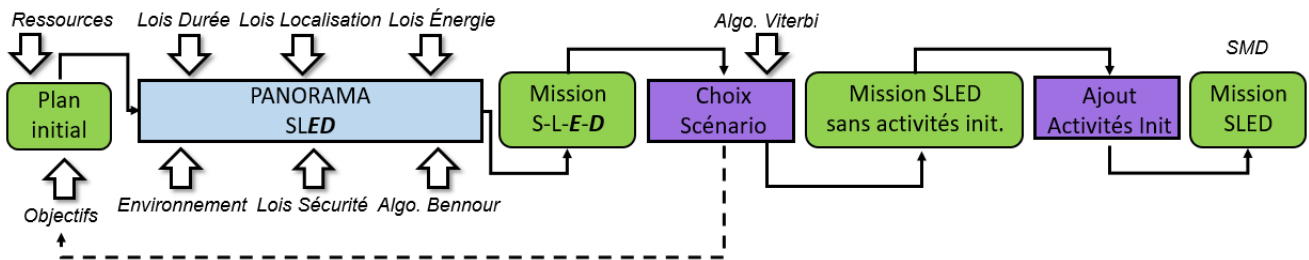


FIGURE 6.8 – PANORAMA(SLED2) : Principales étapes

Elle est constituée des principales étapes méthodologiques suivantes :

1. On commence par fixer des contraintes fin de mission  $C_{init}$  n'intégrant pas le surcoût induit par l'initialisation des méthodes de localisation. On effectue alors une démarche PANORAMA SLED2 qui, au contraire de l'approche SLED1, n'intègre pas ici les coûts d'initialisation. A l'issue de cette étape on obtient, s'il existe, un ensemble d'alternatives, satisfaisant les contraintes cœur de mission (Sécurité et Localisation) mais simplement partiellement les contraintes fin de mission (Energie, Durée).
2. Dans un second temps, on ajoute les surcoûts d'initialisation pour les méthodes de localisation éligibles et on applique l'algorithme de Viterbi.
3. A l'issue du choix de scénario de Viterbi soit la solution trouvée satisfait les contraintes fixées et on a donc obtenu un scénario de Mission. Soit la solution trouvée ne satisfait pas une ou plusieurs des contraintes fin de mission imposées de  $\Delta(C)$ . Dans ce cas on itère à nouveau SLED2 en adoptant comme contrainte initiale fin de mission  $C_{init} = C_{init-1} - \Delta(C)$ .

Les 2 démarches SLED que nous venons de proposer ne s'opposent pas. A contraintes initiales identiques la méthode SLED2 s'avérera plus performante que SLED1. En effet, en offrant la possibilité d'itérer le processus elle tend à préserver l'espace d'états pour l'algorithme de Viterbi alors que l'approche SLED1 aura directement échoué.

En revanche, si l'on admet qu'il est possible de surestimer les contraintes fin de mission initiales pour SLED1, alors les 2 approches ne donneront pas nécessairement les mêmes résultats. L'accroissement du nombre d'alternatives éligibles de SLED1 conduira donc a priori à une solution de meilleure qualité puisque l'algorithme de Viterbi s'appliquera sur un treillis comportant plus de nœuds. Au contraire, SLED2, qui fixe des contraintes d'initialisation plus proches de la réalité ressentie par l'utilisateur, conduit inévitablement à limiter le nombre d'alternatives éligibles et donc potentiellement à appauvrir la qualité de la solution obtenue après l'application de l'algorithme de Viterbi.

On peut alors s'interroger sur la pertinence du filtrage par rapport aux contraintes fin de mission lorsque l'on utilise l'algorithme de Viterbi. Nous reviendrons sur ce point dans la conclusion générale de ce document.

De façon à bien comprendre les différences entre le traitement SED et SLED, un exemple didactique va maintenant être exposé.

### 6.4.3 Exemple didactique

Sans perdre en généralité nous n'évoquerons dans la suite que la sélection des méthodes de localisation et l'énergie correspondante. Elle reste donc compatible avec les 2 alternatives que nous venons d'évoquer. Dans le cas général cela se rapporterait, au niveau d'une activité, à la sélection d'une alternative d'implémentation intégrant toutes les ressources logicielles et matérielles impliquées. Évidemment dans l'exemple traité toutes les valeurs énergétiques employées sont arbitraires. Pour simplifier l'obtention du scénario final de PANORAMA(SLED) nous admettrons que le délais de préparation des méthodes de localisation est inférieur à la durée de l'activité précédent son emploi.

Supposons que dans la mission qui nous intéresse nous ne disposions au maximum que de 7.6 Wh et que cette valeur constitue donc notre objectif de performance énergétique.

Considérons tout d'abord le cas de la mise en œuvre de PANORAMA(SED). La Figure 6.9 présente la situation rencontrée après l'application des contraintes initiales générales, sécuritaires, de Durée et spatiales liées aux méthodes de localisation. Le scénario initial de la Mission est alors divisé en 4 activités (A1, .., A4) correspondant à autant de colonnes. On trouve alors horizontalement les méthodes de localisation (M1 à M3) qui ici peuvent toutes s'appliquer au niveau des différentes activités. Rappelons que ce dernier filtrage, lorsqu'il fait disparaître des méthodes ne s'appuie, pour la méthodologie SED, que sur une expertise utilisateur empirique. Les arcs représentent les enchainements possibles de méthodes d'activité en activité. Enfin les cercles correspondent au coût énergétique estimé de la méthode considérée pour une activité donnée. Aucun coût de préparation n'est porté sur les arcs de liaison. Ainsi, pour l'activité A1, la méthode M1 correspond à un coût énergétique de 5.1 Wh. Remarquons encore qu'au niveau de chaque activité les coûts énergétiques ont été ordonnés.

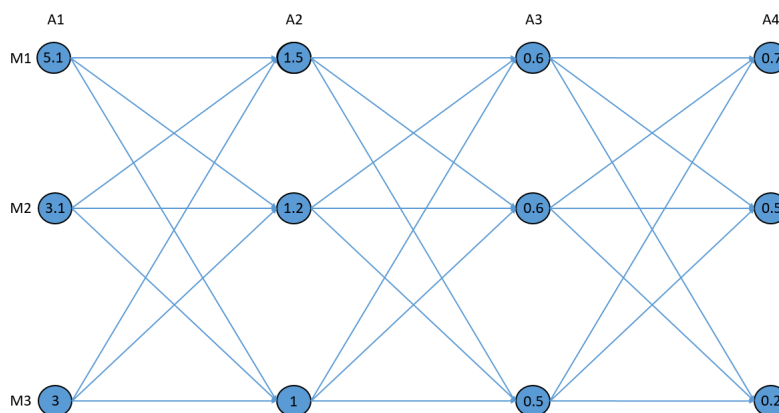


FIGURE 6.9 – PANORAMA(SED) : Situation initiale

L'utilisation de l'algorithme de Bennour va nous permettre de trouver une bon enchainement de méthodes, maximisant l'énergie consommée tout en satisfaisant la contrainte de consommation imposée. Comme le montre la Figure 6.10, le scénario trouvé (cercles et arcs

verts) sélectionne les méthodes M1, M1, M1 et M3 au niveau des activités A1, A2, A3, et A4 respectivement. On trouve en blanc sur fond vert la valeur estimée de la consommation énergétique, au niveau de chaque activité, depuis le lancement de la mission. On constate donc que finalement le SMD déterminé conduit à une consommation théorique de 7.4 Wh et donc à une marge d'énergie de 0.2 Wh.

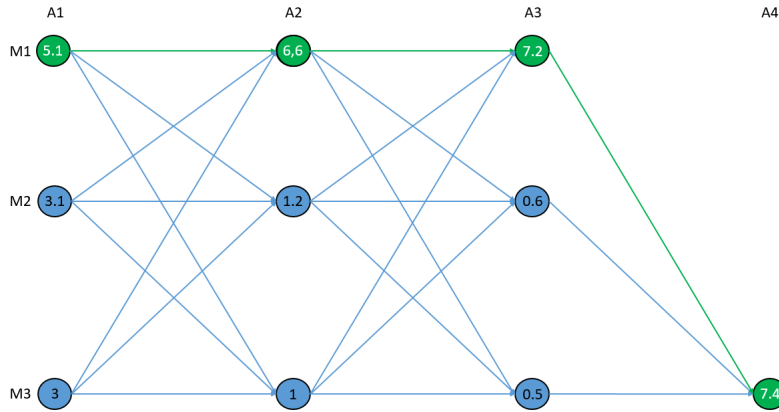


FIGURE 6.10 – PANORAMA(SED) : Scénario SMD obtenu

Considérons à présent le cas de la méthodologie PANORAMA(SLED). La Figure 6.11 part de la même situation initiale que pour PANORAMA(SED) et utilise les mêmes notations et correspondances. Cependant on a fait figurer sur les arcs liant 2 méthodes le coût énergétique de préparation de la méthode cible. Ainsi le passage de la méthode M2 dans l'activité A1 à la méthode M3 dans l'activité A3 induit un coût de préparation de 0.7 Wh. Bien évidemment le passage de A1 à A2 en préservant la sélection de M2 n'induit aucun coût de préparation. L'ajout de ces coûts inter-méthodes constitue une première différence avec le traitement de la localisation dans l'approche PANORAMA(SED). On observe bien évidemment l'adéquation de la structure en treillis au problème de sélection des méthodes de Localisation qui nous intéresse.

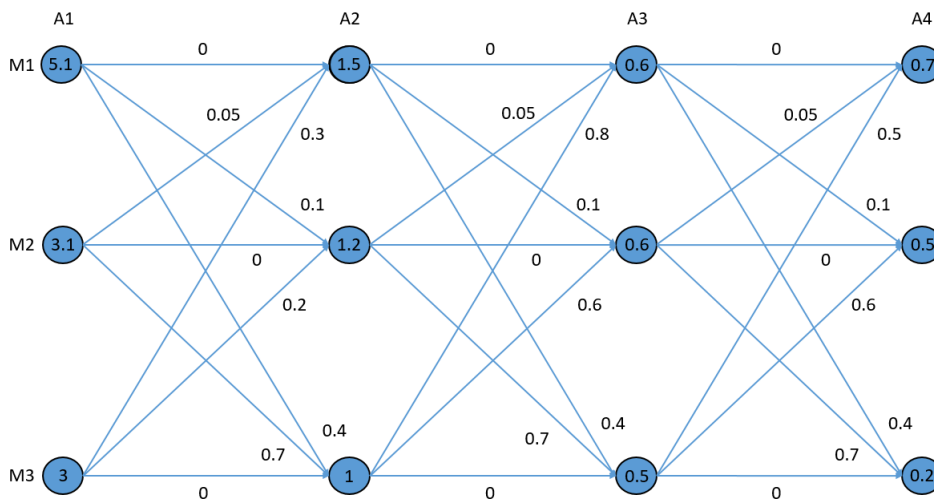


FIGURE 6.11 – PANORAMA(SLED) : Situation initiale

La Figure 6.12 présente maintenant la structure du treillis après vérification des contraintes de Localisation imposées par l'utilisateur grâce à l'emploi de nos modèles prédictifs. On constate que 3 noeuds ont disparu du treillis. M1 pour A1, M2 pour A3 et enfin M3 pour A4. Ainsi nous

ne préservons que les méthodes dont l'utilisation et l'enchaînement peuvent permettre effectivement de satisfaire les contraintes de pose imposées. Dès à présent on peut conclure que le scénario proposé par PANORAMA(SED) viole de toute évidence les contraintes de pose en raison de l'utilisation de la méthode M3 dans l'activité A4. Cela démontre donc que la prise en compte qualitative des méthodes de localisation effectuée pour une garantie SED est imparfaite.

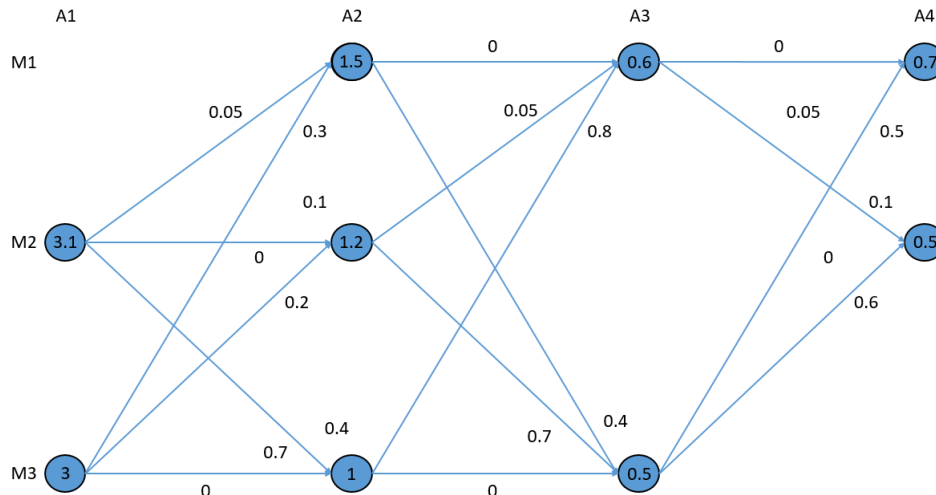


FIGURE 6.12 – PANORAMA(SLED) : Filtrage garantissant la Localisation

La mise en œuvre de l'algorithme de Viterbi sur le treillis précédent conduit à la Figure 6.13. Pour chaque activité, au niveau de chaque méthode, on trouve en blanc sur fond vert ou bleu le coût énergétique du chemin minimal menant au nœud considéré. Les cercles verts matérialisent les méthodes constituant le scénario de la Mission dont le coût énergétique minimal est évalué à 5.55 Wh. Plusieurs remarques peuvent être effectuées sur la nature de ce dernier. D'une part évidemment, ce ne sont pas toujours les méthodes les plus énergivores qui ont été sélectionnées localement contrairement au postulat avancé dans PANORAMA(SED). D'autre part le maintien d'une méthode de localisation entre 2 activités n'est pas particulièrement privilégié. C'est le cas entre A1 et A2 où M2 est préservée (arc vert), mais la fonction de coût énergétique conduit à effectuer 2 commutations inter-méthodes pour atteindre A4 (arcs rouges).

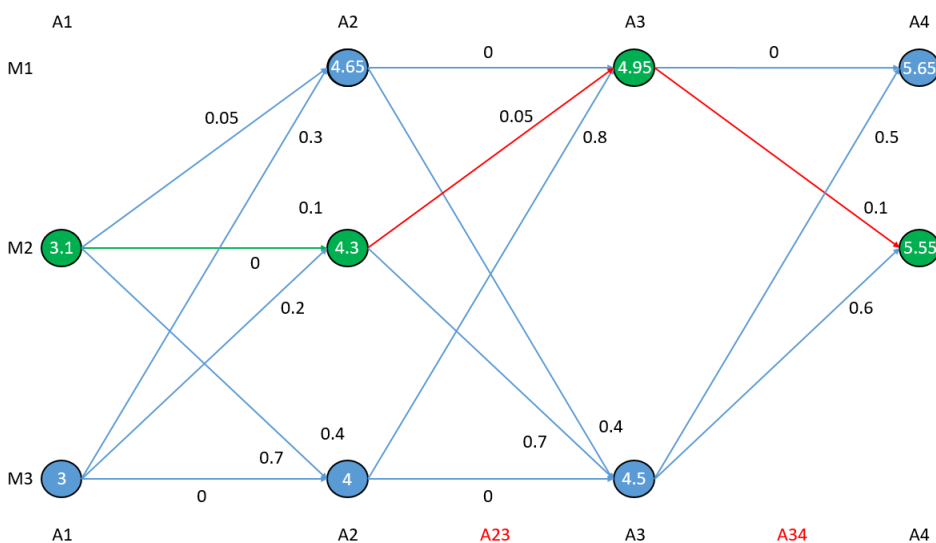


FIGURE 6.13 – PANORAMA(SLED) : Scénario SMD obtenu

Il reste la dernière étape de la méthode PANORAM(SLED) à déployer pour reconstituer le scénario final SMD en partant de l'état final d'énergie minimale. On ajoute donc au niveau du scénario initial les activités de transition A23 et A34 nécessaires à la préparation des méthodes de localisation lorsqu'une commutation est nécessaire. Le scénario de départ composé de 4 activités conduit donc finalement à un SMD de 6 activités.

Notons pour terminer qu'évidemment la marge d'énergie de 0.2 Wh obtenue pour la PANORAM(SED) a été très fortement amplifiée pour atteindre 2.05 Wh en utilisant l'algorithme de Viterbi pour PANORAMA(SLED).

## 6.5 Conclusion

Ce chapitre a permis dans un premier temps de détailler les spécificités de l'axe de performance Localisation et leurs impacts sur la méthodologie PANORAMA. Tout d'abord, contrairement aux autres axes de performance cet axe présente une corrélation inter-activités induite par les délais de préparation d'une méthode de localisation. Cela impacte nécessairement l'estimation de la consommation énergétique et donc la marge correspondante. Enfin, contrairement aux autres axes de performance nous ne disposons pas d'un observateur externe permettant en permanence, lors de l'exécution de la mission, de connaître la valeur réelle courante de l'indicateur de performance (Energie consommée, durée actuelle de la mission) et de réajuster en conséquence la valeur estimée de la marge de performance correspondante.

Dans un second temps nous avons présenté comment l'algorithme de Viterbi, pouvant être utilisé en programmation dynamique, pour déterminer un chemin à coût minimal au sein d'un treillis, est parfaitement adapté pour sélectionner les méthodes de localisation devant être employées tout au long d'une mission pour satisfaire les contraintes de localisation imposées. Qui plus est, il nous permet de maximiser la marge énergétique disponible en début de mission. Cela constitue donc l'un des résultats majeurs de ce travail.

Loin d'être rendus caduques les travaux présentés dans [112], à l'origine de la méthodologie PANORAMA(SED) et permettant une garantie SED, sont complétés par les présents résultats et on conduit à proposer PANORAMA(SLED) assurant une garantie SLED. Enfin, pour saisir les différences de comportement de ces deux déclinaisons de la méthodologie PANORAMA un exemple didactique a été détaillé.

Il nous reste maintenant à mettre à l'épreuve de l'expérimentation la méthodologie PANORAMA(SLED) et la garantie SLED qu'elle doit procurer sur la Mission de patrouille qui nous sert de fil rouge.

# Chapitre 7

## VERS UNE GARANTIE DE PERFORMANCE INCLUANT LA LOCALISATION : SIMULATIONS ET EXPÉRIMENTATIONS

Ce chapitre va permettre de détailler la nouvelle démarche PANORAMA(SLED) de construction des Scénarios de Mission Détaillée et d'analyser la pertinence des modèles d'incertitude de localisation développés. Il se décompose en 4 principales parties. La première décrit la mission étudiée au travers de l'ensemble des contraintes à satisfaire et précise les objectifs de performance recherchés. La seconde, présente rapidement l'outil de simulation utilisé. Dans la troisième nous nous intéressons aux résultats obtenus en ne faisant appel qu'aux seuls amers verticaux. Nous les analyserons au travers de la simulation et de l'expérimentation. Pour terminer nous ferons appel aux amers horizontaux pour réaliser pleinement la mission fixée. Après en avoir présenté la simulation, les résultats expérimentaux obtenus sont exposés d'une part en détaillant le déroulement d'une mission particulière et, d'autre part, en analysant plus globalement l'ensemble des expérimentations réalisées.

### 7.1 Mission étudiée : Hypothèses, contraintes et objectifs

Après un rappel de la mission étudiée, nous présentons ici, de façon assez synthétique, les schémas de contrôle pouvant être mis en œuvre ainsi que les hypothèses de travail retenues, les contraintes prises en compte, et pour finir les objectifs de performance adoptés pour réaliser la mission de Patrouille que nous avons choisie comme fil rouge.

#### 7.1.1 Mission de Patrouille : Description initiale

La mission de Patrouille qui nous intéresse a pour objectif de parcourir 2 couloirs du sous-sol du laboratoire pour inspecter l'état (ouvert/fermé) de 2 vannes, puis de revenir à son point de départ.

Rappelons ici que cette mission se décompose en un ensemble d'**objectifs** devant séquentiellement être exécuté. Chacun d'eux nécessitant la mise en œuvre concurrente d'une ou plusieurs **Tâches Robotiques**.

Ainsi la mission Patrouiller peut se décomposer en une séquence d'objectifs (Table 7.1) faisant appel à une ou plusieurs tâches robotiques.



Objectif	Description	Tâches robotiques
O1	Aller à V1	DEP / LOC
O2	S'orienter vers V1	ROT / LOC
O3	Inspecter l'état de V1	ANIM
O4	S'orienter vers le chemin	ROT / LOC
O5	Aller à V2	DEP / LOC
O6	S'orienter vers V2	ROT / LOC
O7	Inspecter l'état de V2	ANIM
O8	Se retourner	ROT / LOC
O9	Aller à D	DEP / LOC

DEP : DEPlacement – LOC : LOCALisation – ROT : ROTation – ANIM : ANalyse IMage

TABLE 7.1 – Objectifs de la mission de Patrouille

Les tâches robotiques ayant été identifiées nous allons maintenant détailler les différents schémas de contrôle et ressources pouvant être mobilisés pour les mettre en œuvre.

### 7.1.2 Tâches robotiques : schémas de contrôle, ressources matérielles et logicielles

La mise en œuvre d'une mission suppose évidemment l'utilisation d'un ensemble de schémas de contrôle ainsi que l'adoption d'hypothèses de travail et la prise en compte de contraintes qui vont maintenant être résumés.

### 7.1.3 Schémas de contrôle et ressources associées

Pour exécuter la mission de Patrouille qui nous intéresse nous disposons des schémas de contrôle présentés dans la Table 7.2.

Pour chacune des tâches robotiques on y indique le rôle associé ainsi que les différentes combinaisons de capteurs et d'actionneur pouvant y être associées (●). On trouve dans la dernière colonne le référencement qui sera utilisé dans la suite de ce document pour identifier les schémas sélectionnés par PANORAMA sachant qu'en présence de doublet le second élément désignera la tâche de Localisation. Ainsi (1/2) désigne DEP avec sonars et moteurs, et LOC avec odométrie, alors que (7) correspond simplement à ANIM avec caméra et moteurs allumés.

Passons à présent à la présentation des hypothèses de travail qui ont été adoptées.

### 7.1.4 Hypothèses de travail

Les hypothèses de travail de la Table 7.3 ont été retenues pour la mission de Patrouille.

On suppose évidemment que le robot est matériellement adapté à la mission visée et que les schémas de contrôle dont il dispose le sont aussi. La connaissance de la quantité d'énergie disponible est indispensable pour pouvoir déterminer si la mission est réalisable en l'état. Les modèles prédictifs de Durée et Énergie nécessitent de connaître le chemin devant être théoriquement suivi par le robot et son environnement. Même si dans notre cas nous le considérerons comme étant une succession de lignes droites, rien ne s'oppose dans la démarche proposée, à prendre des chemins plus complexes. Enfin on suppose que la fréquence d'échantillonnage permet d'assurer la stabilité du contrôle.

Tâche Robotique	Logiciel		Matériel					Réf
	Type	Info.	Sonar	Laser	Kinect 1	Kinect 2	Actionneur	
DEP	Guidage & Contrôle	<i>Suivi de chemin avec évitement d'obstacle</i>	•	-	-	-	•	1
			-	•	-	-	•	2
			•	•	-	-	•	3
			-	••	-	-	•	4
			•	••	-	-	•	5
ROT	Guidage & Contrôle	<i>Rotation</i>	-	-	-	-	•	6
ANIM	Analyse d'image	<i>Détection état d'une vanne</i>	-	-	•	-	•	7
LOC	Localisation	ODO	-	-	-	-	•	1
		KINODO	-	-	•	-	•	2
		COMPODOKIN	-	••	•	-	•	3
		LZA-ODO	-	••	-	-	•	4
		LZA-ODO-COMPAS	-	••	-	-	•	5
		KIN2ODO	-	-	-	•	•	6

TABLE 7.2 – Schémas de contrôle et ressources associées

Hypothèses	Description
Hyp. 1	L'environnement où s'exécute la mission est connu
Hyp. 2	Les capacités de perception et de déplacement du robot sont compatibles avec la mission
Hyp. 3	Les algorithmes disponibles (contrôle, localisation, etc.) sont compatibles avec la mission
Hyp. 4	La quantité minimale d'énergie embarquée disponible du robot est connue
Hyp. 5	La stabilité du contrôle du robot peut être assurée avec une période d'échantillonnage de 10 Hz
Hyp. 6	Le chemin à suivre durant la mission est connu

TABLE 7.3 – Hypothèses associées à la mission de Patrouille

Le cadre général de la mission étant tracé, abordons à présent la présentation des contraintes qui devront être considérées.

### 7.1.5 Contraintes

Nous distinguerons les contraintes matérielles, des contraintes expérimentales, de celles associées aux méthodes de localisation et enfin des contraintes de performance imposées par l'utilisateur. Nous allons maintenant les balayer.

### 7.1.5.1 Contraintes matérielles et expérimentales

La Table 7.4 résume les contraintes matérielles et expérimentales de la mission de Patrouille.

Contrainte	Localisation (m) ( $x_{deb} - x_{fin}$ )	Description
$C1_{Mat}$	(0-178.3)	La vitesse maximale du robot est de $v_{max}(Robot) = 0.76 \text{ m/s}$
$C1_{Exp}$	(0-178.3)	Des êtres Humains peuvent être présents tout au long de la mission
$C2_{Exp}$	(25-35) (143.3-153.3)	Il y a 2 zones vitrées dans l'environnement d'expérimentation
$C3_{Exp}$	(0-6)(83-95.3) (172.3-178.3)	Zone avec meubles bas non détectés par les lasers
$C4_{Exp}$	(64.2-67) (111.3-114.1)	lasers non utilisables (texture, couleur)
$C5_{Exp}$	(0-178.3)	La vitesse maximale du robot est limitée à $v_{max}(Kinect) = 0.5 \text{ m/s}$ lorsque qu'une Kinect est utilisée

TABLE 7.4 – Contraintes matérielles et expérimentales de la mission de Patrouille

Les contraintes présentées sont liées aux conditions d'utilisation des capteurs, à l'environnement dynamique de la mission et aux caractéristiques du robot ( $v_{max}(Robot)$ ).

Passons maintenant en revue les contraintes d'utilisation associées aux méthodes de localisation.

### 7.1.5.2 Contraintes d'utilisation des méthodes de localisation

Nous terminerons notre description préalable du contexte de la mission de Patrouille par la définition des contraintes de performance qui correspondront aux objectifs de performances que nous chercherons à satisfaire simultanément grâce au choix judicieux d'allocation et de paramétrage des schémas de contrôle éligibles.

### 7.1.5.3 Contraintes/objectifs de performance

La Table 7.5 résume les domaines d'application des contraintes de performance en fonction de leur nature, cœur de mission (Sécurité, Localisation) ou fin de mission (Durée, Énergie).

Contrainte	Localisation (m) ( $x_{deb} - x_{fin}$ ) ou $x_{loc}$	Description
$C1_{Perf}$	(0-178.3)	La Sécurité doit être assurée tout au long de la mission
$C2_{Perf}$	(0-178.3)	La Localisation doit être assurée tout au long de la mission
$C3_{Perf}$	178.3	Durée : La mission ne doit pas dépasser $C3_{Perf} \text{ s}$
$C4_{Perf}$	178.3	Énergie : La mission ne doit pas consommer plus de $C4_{Perf} \text{ Wh}$

TABLE 7.5 – Espaces de garantie de performance de la mission de Patrouille

Si l'on explicite maintenant la valeur des différentes contraintes et de leurs domaines d'application qui seront appliqués par la suite pour élaborer les Scénarios de Mission Détaillés, on peut construire la Table 7.6.

Axe de Performance	Localisation (m) ( $x_{deb} - x_{fin}$ ) ou $x_{loc}$	Valeurs (m, m, rad) ( $X_s, Y_s, \theta$ )	Description
Sécurité	-	$v_{max}(US) = 0.33 \text{ m/s}$	Vitesse maximale si les sonars sont utilisés
Sécurité	-	$v_{max}(Laser) = 0.4 \text{ m/s}$ $v_{max}(Laser, US) = 0.4 \text{ m/s}$	Vitesse maximale si les lasers sont utilisés
Localisation	(0-6) (172.3-178.3)	(0.8, 0.8, $\pi/4$ )	<b>L1</b> - Traversée de la halle
Localisation	(6-25.5) (31.5-57) (121.3-172.3)	(1, 0.85, $\pi/4$ )	<b>L2</b> - Traversée du couloir 1
Localisation	(25.5-31.5) (85.5-89.15)	(0.4, 0.8, $\pi/4$ )	<b>L3</b> - Approche vannes
Localisation	(57-60) (118.3-121.3)	(0.5, 1, $\pi/4$ )	<b>L4</b> - Angle entre couloirs 1 et 2
Localisation	(60-83) (95.3-118.3)	(1, 1, $\pi/4$ )	<b>L5</b> - Traversée couloir 2
Localisation	(83-85.5) (89.5-95.3)	(0.8, 0.8, $\pi/4$ )	<b>L6</b> - Zone de repos
Durée	178.3	Durée Mission $\leq 600 \text{ s}$	Durée maximale de la mission
Énergie	178.3	Énergie robot $\leq 2.4 \text{ Wh}$	Énergie maximale que le robot peut consommer

TABLE 7.6 – Objectifs de performance à garantir pour la mission de Patrouille

Les contraintes sécuritaires se traduisent au travers de la définition de vitesses maximales atteignables pour assurer l'innocuité d'un choc ( $Energie_{Choc} \leq 4J$ ), en fonction des capteurs proximométriques utilisés et du rayon de sécurité adopté ( $r_s = 0.65 \text{ m}$ ) pour la méthode d'évitement d'obstacle utilisée [143]. Les pavés garantis pour l'axe Localisation varient en fonction des zones traversées. On améliore la précision demandée en diminuant l'incertitude tolérée au voisinage des vannes. Les valeurs affichées peuvent sembler importantes mais il ne faut pas oublier que nous sommes dans une logique de garantie et non pas dans une logique probabiliste. Enfin les limites affichées pour les axes de Durée et Énergie indiquent d'une part la durée maximale d'exécution de la mission de Patrouille et, d'autre part, l'énergie maximale qu'il est possible de dépenser pour la réaliser.

Les contraintes affectant le déroulement de notre mission fil rouge sont maintenant fixées. Nous allons maintenant rapidement présenter le simulateur développé pour déployer l'approche, de l'approche PANORAMA.

## 7.2 Présentation du simulateur

Cette partie présente le simulateur développé pour construire les Scénarios de Mission Détaillé (SMD) en fonction des contraintes de performance spécifiées en faisant appel aux stratégies SED et SLED2 de la méthodologie PANORAMA.

Le simulateur développé reprend très largement celui présenté dans [112]. Il permet à partir d'une description textuelle de la mission de mettre en œuvre chacune des stratégies d'allocation des ressources matérielles et logicielles élaborées. On obtient, pour chacune des activités de la mission, les schémas de contrôle sélectionnés ainsi que leur paramétrage (vitesse, fréquence, etc.). De plus les marges de performance "théoriques" peuvent aussi être tracées. Afin de rajouter plus de réalisme il est possible d'introduire des évitements d'obstacles faisant dévier le robot de la trajectoire rectiligne de base, ou de simuler des fautes capteurs ou logicielles.

Les paramètres du simulateur sont saisis via une interface constituée de 6 panneaux principaux (Figure 7.1) :

1. Le panneau **Mission** permet principalement de charger la description textuelle de la mission mémorisée sous la forme d'un fichier, de spécifier la durée de la mission et de sélectionner la stratégie d'allocation de ressources à employer (SED, SLED1 ou SLED2).
2. Le panneau **Contraintes énergétiques** permet de préciser les contraintes de consommation énergétique imposées pour la mission sur les batteries robot et laptop. Cette dernière n'est pas considérée dans le travail présenté.
3. Le panneau **Évènements** permet d'introduire des évènements d'obstacles générés aléatoirement selon une distribution choisie (loi Uniforme, Gaussienne ou Exponentielle), ou de sélectionner explicitement leur emplacement. De la même façon il est aussi possible de simuler des fautes sur des ressources particulières. Nous ne ferons pas appel à ce générateur d'évènements dans ce document de façon à comparer les marges énergétiques brutes obtenues par chacune des stratégies d'allocation de ressources.
4. Le panneau **Paramètres** permet de renseigner les paramètres structurels du robot (masse, vitesse maximale) ainsi que la dimension du rayon de sécurité choisie pour l'évitement d'obstacle.
5. Le panneau **Simulation** permet de définir le nombre de simulations à réaliser dans le cas d'une génération aléatoire d'évènements, et d'effectuer les simulations correspondantes.
6. Le panneau **Résultats** permet d'afficher les résultats que l'on souhaite analyser : Marges de performance, déroulement de la mission et variation de vitesse.

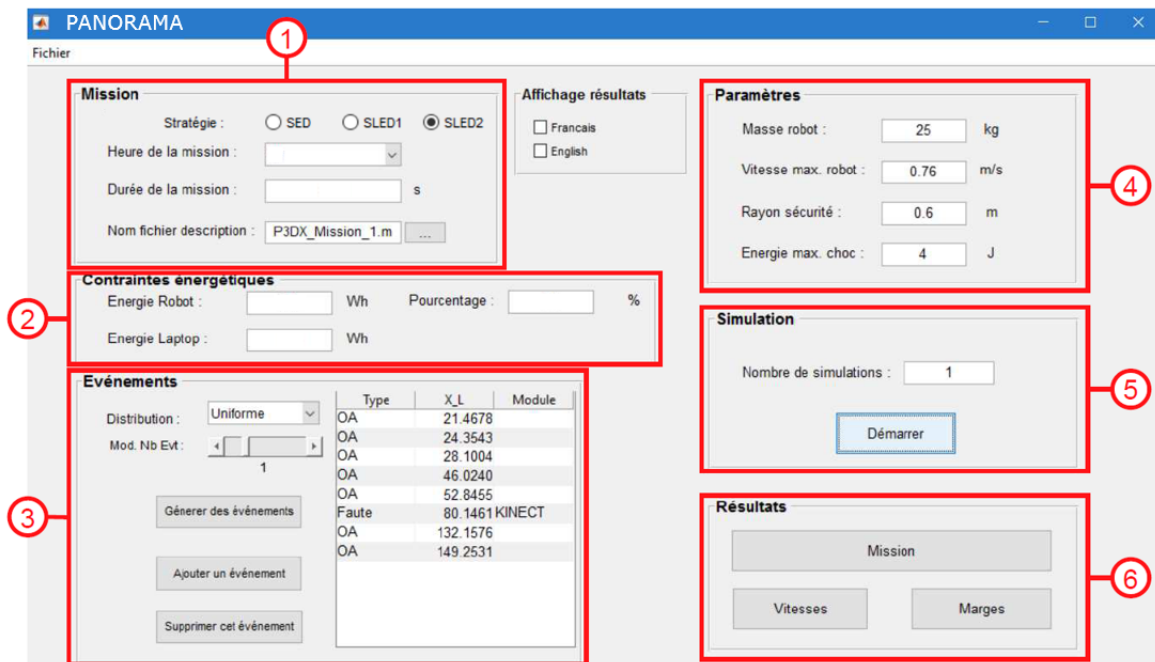


FIGURE 7.1 – Simulateur PANORAMA : Interface graphique

Passons maintenant en détail la mise en œuvre des stratégies PANORAMA sur la mission de Patrouille qui nous intéresse.

### 7.3 Utilisation des QR-Codes verticaux

Dans cette partie nous allons chercher à réaliser la mission fixée en ne faisant appel qu'aux uniques marqueurs verticaux pour recalculer la localisation.

### 7.3.1 PANORAMA(SED) : Scénario de Mission Détaillé

Nous ne reprendrons pas ici en détail la genèse de construction du Scénario de Mission Détaillé décrit dans [112] et dont le résultat est synthétisé au travers des Tables 7.7 et 7.8 décrivant l’aller puis le retour de la mission de Patrouille étudiée.

Dans ces tables on trouve les informations suivantes :

Les objectifs, activités et contraintes sont définis sur des zones qui sont délimitées spatialement par leurs coordonnées initiales et finales. Celles-ci représentent, par rapport au début de la mission, les coordonnées linéaires de début et de fin de la zone considérée. Selon la vitesse adoptée, une dualité distance/temps peut être déduite.

- Dans ces tables les objectifs, contraintes et activités sont délimités spatialement, en colonnes, par leur abscisse linéaire de début et de fin d’exécution. Celles-ci correspondent au chemin parcouru par le robot depuis sa position initiale.
- La première ligne renseigne la succession des objectifs de la mission ainsi que les zones d’exécution correspondant à des doublets  $(x_{deb}, x_{fin})$ . On notera qu’au niveau des vannes, le robot ne bougeant pas linéairement, la zone est réduite à une seule abscisse.
- La seconde ligne indique les tâches robotiques devant être mises en œuvre pour réaliser un objectif.
- La ligne informations recouvre les données suivantes : Abscisse de début de l’activité  $x_{AI}$ , longueur de l’activité  $dis_{AI}$  ainsi que sa durée  $dur_{AI}$  si celle-ci est indépendante du paramétrage du schéma de contrôle.
- Les 2 ensembles de lignes suivants indiquent, en grisé, les zones d’application des contraintes expérimentales et matérielles.
- Le groupe de lignes suivant reporte en rouge les zones d’éligibilité des différentes méthodes de localisation étudiées repérées selon le référencement adopté dans la Table 7.2.
- Le groupe suivant de lignes est associé à la prise en compte des contraintes sécuritaires. Cela induit l’utilisation de vitesses de déplacement variables en fonction de l’environnement et des capteurs impliqués. Ainsi on trouve en vert les zones où la contrainte d’innocuité s’applique, et pour l’évitement d’obstacle (EO) en bleu celles où les sonars doivent impérativement être employés (zones vitrées) et enfin en jaune celles où la vitesse sera limitée en fonction des capteurs extéroceptifs retenus par PANORAMA (sonars et/ou laser).
- La projection de l’ensemble de ces contraintes permet d’identifier les activités qui constitueront le SMD.
- La ligne  $n_{AI}$  désigne, pour chaque activité, le nombre d’Alternatives d’Implémentation possible parmi lesquelles PANORAMA devra sélectionner et paramétrer celle qui permettra globalement de satisfaire les objectifs de performance imposés.
- La dernière ligne indique, pour chaque activité, les ressources matérielles et logicielles sélectionnées en appliquant la stratégie PANORAMA(SED). Cela correspond à la Solution d’Allocation de Ressources initiale  $SAR_0$  qui est déterminée hors-ligne.

Objectif ( $X_{deb} - X_{fin}$ )	$O_I$ (0 – 31.5)						$O_{II}$ (31.5) ROT   LOC	$O_{III}$ (31.5) ANIM	$O_{IV}$ (31.5) ROT   LOC	$O_V$ (31.5 – 89.15)										
	DEP   LOC									DEP   LOC										
Tâches Robotiques	0	6	9	12	25	31.5	31.5	31.5	31.5	31.5	35	57	60	64	64.2	67	69	73	79	83
Informations	$X_{AI}$	6	3	3	13	6.5	0	10	0	0	22	3	4	0.2	2.8	2	4	6	4	6.15
	$dis_{AI}$	-	-	-	-	-	10	10	1	10	-	-	-	-	-	-	-	-	-	-
	$dur_{AI}$	-	-	-	-	-	10	10	1	10	-	-	-	-	-	-	-	-	-	-
Contraintes Expérimentales	$C_{1EXP}$																			
	$C_{2EXP}$																			
	$C_{3EXP}$																			
	$C_{4EXP}$																			
	$C_{5EXP}$																			
Contrainte matérielle	$C_{1MAT}$																			
Méthode de localisation possible	1																			
	2																			
	3																			
	4																			
	5																			
Contraintes de sécurité	EO																			
	Inno cuité																			
Activité	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$	$A_8$	$A_9$	$A_{10}$	$A_{11}$	$A_{12}$	$A_{13}$	$A_{14}$	$A_{15}$	$A_{16}$	$A_{17}$	$A_{18}$	$A_{19}$	
$n_{AI}$	6	10	15	25	15	5	1	5	15	25	15	15	25	25	25	15	25	15	6	
$SAR_0$	5/2	5/2	1/3	1/3	1/3	6/3	7	6/3	1/3	1/3	5/2	1/3	1/3	1/3	1/3	1/3	1/3	5/2	5/2	

**Code couleur des activités**

	Robot uniquement en rotation		Robot immobile		$V_{max} = 0.33$ m/s		$V_{max}$ fonction des capteurs utilisés
--	------------------------------	--	----------------	--	----------------------	--	--

TABLE 7.7 – PANORAMA SED : Scénario de Mission Détaillé et Solution d’Allocation de Ressources - Partie 1 - Aller

Objectif ( $X_{deb} - X_{fin}$ )		O <sub>IX</sub> (89.15 – 178.3)																	
Tâches Robotiques		DEP   LOC																	
		O <sub>VI</sub> (89.15) ROT   LOC	O <sub>VII</sub> (89.15) ANIM	O <sub>VIII</sub> (89.15) ROT   LOC	95.3	99.3	103.3	109.3	111.3	114.1	114.3	118.3	121.3	143.3	153.3	166.3	169.3	172.3	
Informations	X <sub>AI</sub>	89.15	89.15	89.15	95.3	99.3	103.3	109.3	111.3	114.1	114.3	118.3	121.3	143.3	153.3	166.3	169.3	172.3	
	dis <sub>AI</sub>	0	0	0	4	4	6	2	2.8	0.2	4	3	22	10	13	3	3	6	
	dur <sub>AI</sub>	10	1	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	C <sub>1EXP</sub>																		
	C <sub>2EXP</sub>																		
Contraintes Expérimentales	C <sub>3EXP</sub>																		
	C <sub>4EXP</sub>																		
	C <sub>5EXP</sub>																		
	C <sub>1Mat</sub>																		
Contrainte matérielle	1																		
	2																		
	3																		
	4																		
	5																		
Méthode de localisation possible	EO																		
	Inno																		
	cuité																		
Contraintes de sécurité	A <sub>20</sub>	2	A <sub>20</sub>	A <sub>23</sub>	A <sub>24</sub>	A <sub>25</sub>	A <sub>26</sub>	A <sub>27</sub>	A <sub>28</sub>	A <sub>29</sub>	A <sub>30</sub>	A <sub>31</sub>	A <sub>32</sub>	A <sub>33</sub>	A <sub>34</sub>	A <sub>35</sub>	A <sub>36</sub>	A <sub>37</sub>	
	n <sub>AI</sub>	6/2	1	2	9	25	15	25	15	25	15	15	25	15	25	15	10	6	
	SAR <sub>0</sub>	6/2	7	6/2	5/2	1/3	1/3	1/3	1/3	1/3	1/3	5/2	1/3	3/2	1/1	1/1	1/1	1/1	

**Code couleur des activités**

	Robot uniquement en rotation		Robot immobile		$V_{max} = 0.33 \text{ m/s}$		$V_{max}$ fonction des capteurs utilisés
--	------------------------------	--	----------------	--	------------------------------	--	--

TABLE 7.8 – PANORAMA SED : Scénario de Mission Détaillé et Solution d’Allocation de Ressources - Partie 2 - Retour



Dans ces tables évidemment, pour la stratégie SED, les contraintes de localisation ne prennent pas en compte les distances nécessaires à leur initialisation.

La projection des contraintes le long de la mission conduit donc à la création de  $nb_{act} = 37$  activités à contraintes constantes et de dimensions variables allant de 0.2 m à 22 m.

La combinatoire du problème d'allocation de ressources abordé est égale au produit du nombre d'alternatives d'implémentation de chaque activité (Équation 7.1). Sa valeur démontre l'efficacité de la démarche PANORAMA pour trouver de bonnes solutions d'allocation au sein d'un espace de recherche réellement très important.

$$Complexité(SED) = \sum_{i=1}^{nb_{act}} n_{Ai} = 5.4 \cdot 10^{38} \quad (7.1)$$

En considérant les contraintes de performance choisies dans la Table 7.6 la stratégie PANORAMA SED sélectionne donc les schémas de contrôle de la dernière ligne des Tableaux 7.7 et 7.8. On peut alors compléter la Table 7.9. Elle identifie les schémas de contrôle employés et comptabilise leur longueur d'utilisation ainsi que leur impact énergétique linéaire et au sein de la mission (%Mission et %Energie).

Schéma de contrôle	DEP (Ressources matérielles)	LOC	Longueur utilisation (m)	% Longueur Mission	%Énergie
(1/1)	SCH-US (US)	ODO	25	14	6.67
(1/3)	SCH-US (US LAS1-2 - Kinect)	COMPODOKIN	108	60.57	60.34
(3/2)	SCH-LAS1-US (US LAS1 - Kinect)	KINODO	10	5.6	4.76
(5/2)	SCH-LAS12-US (US LAS1-2 Kinect)	KINODO	35.3	19.80	18.55
(6/2)	ROT-ODO (Kinect)	KINODO	0	0	3.34
(6/3)	ROT-ODO (LAS1-2 - Kinect)	COMPODOKIN	0	0	4.43
(7)	ANIM (Kinect)	/	0	0	1.88

TABLE 7.9 – PANORAMA(SED) : Utilisation des schémas de contrôle

Pour cette sélection de schémas de contrôle, l'utilisation de nos modèles prédictifs en Durée et Énergie nous permet de déterminer théoriquement la durée de la mission et l'énergie qui doit être consommée. Compte tenu des objectifs de performance imposés on peut donc en déduire les marges de temps et d'énergie théoriques correspondantes (Équations 7.2 et 7.3).

$$M_{TEMPS}(SED)_{th} = Objectif_{Durée} - Durée_{th} = 600 - 573.29 = 26.71 \text{ s} \quad (7.2)$$

$$M_{ENERGIE}(SED)_{th} = Objectif_{Energie} - Energie_{th} = 2.4 - 2.3982 = 0.0018 \text{ Wh} \quad (7.3)$$

On constate que la stratégie PANORAMA(SED) procure une marge de temps significative. En revanche conformément à l'heuristique adoptée maximisant localement la consommation énergétique tout en satisfaisant globalement la contrainte d'énergie imposée, la marge d'énergie obtenue est extrêmement faible. Cela se traduira inévitablement lors de l'expérimentation par l'apparition de marges énergétiques négatives qui conduiront à chercher en cours de mission, toujours grâce à PANORAMA(SED), de nouvelles solutions d'allocation préservant la satisfaction de la contrainte de consommation imposée [112]. Cependant, d'une part, rien n'assure ici que les méthodes de localisation peuvent bien être utilisées, ni que les incertitudes de pose que

nous cherchons à satisfaire pourront effectivement l'être. Par exemple, l'odométrie est choisie pour parcourir les 25 derniers mètres ce qui inévitablement conduira à violer les garanties d'incertitude de localisation que nous chercherons maintenant à satisfaire.

Nous allons donc maintenant détailler la mise en œuvre de la stratégie PANORAMA(SLED2). Elle doit nous permettre de garantir les incertitudes de pose mentionnées tout en maximisant la marge énergétique évitant par la même la remise en cause de la solution d'affectation déterminée à l'exclusion évidemment de la défaillance de certaines des ressources employées. Commençons par envisager de n'utiliser que les seuls QR-Codes de QRC1. Cela exclut de fait l'utilisation de KIN2ODO faisant appel aux marqueurs placés au plafond.

### 7.3.2 PANORAMA(SLED2 - QRC1) : ECHEC

En limitant l'utilisation des QR-Codes au seul ensemble QRC1, la méthodologie PANORAMA(SLED2) ne converge vers aucune Solution d'Allocation de Ressources.

L'analyse de cet échec démontre que les performances d'incertitude de localisation demandées ne peuvent être satisfaites à l'aide des méthodes disponibles et de l'équipement de l'environnement sélectionné au niveau de la salle de repos et au voisinage de la halle d'expérimentation. Cela vient du fait que lorsque la méthodologie globale LZA n'est pas éligible, l'utilisation de KINODO combinant l'odométrie et le recalage par QR-Codes de QRC1 ne permet pas de préserver la garantie de pose demandée en raison d'une distance inter amers trop importante. Par ailleurs la méthode COMPODOKIN n'est pas éligible dans ces zones puisque la forme de l'environnement ne permet pas d'utiliser notre compas logiciel.

Ce résultat confirme bien que la garantie PANORAMA(SED) avancée dans [112] pour la dimension Localisation reposait sur une hypothèse erronée qui présupposait des niveaux de performance d'incertitude de Localisation pour les méthodes sans pour autant apporter de garantie qualitative tangible.

En nous appuyant sur l'analyse de cet échec, nous allons donc enrichir l'instrumentation de notre environnement on y ajoutant les QR-Codes de l'ensemble QRC2 essentiellement au niveau des zones problématiques identifiées.

### 7.3.3 PANORAMA(SLED2 - QRC1 $\cup$ QRC2)

Dans cette partie nous allons tout d'abord présenter la description du Scénario de Mission Détaillé intégrant explicitement les contraintes de localisation. Dans un second temps, avant de conclure, les résultats de la mise en œuvre de l'algorithme de Viterbi avec la stratégie SLED2 seront explicités.

#### 7.3.3.1 PANORAMA(SLED2 - QRC1 $\cup$ QRC2) - SMD : Axes Sécurité / Localisation

Les Tables 7.10, 7.11 et 7.12 synthétisent comme pour PANORAMA(SED) la projection de l'ensemble des contraintes affectant la mission de Patrouille sur le chemin à parcourir. Les mêmes conventions de notation, de couleur ont été adoptées. Il faut cependant souligner que les zones d'éligibilité de la méthode LZA ont été volontairement amputées de 2 m de part et d'autre afin de pouvoir assurer son initialisation. Les portées spatiales des contraintes d'incertitude de Localisation imposées par l'utilisateur (Table 7.6) y ont été précisées. Ces modifications

induisent une décomposition supplémentaire de la mission par rapport à la stratégie SED portant le nombre d'activités du SMD à  $n_{act} = 45$  avec des dimensions comprises entre 0.5 m à 20 m.

Avant la prise en compte des axes Énergie et Localisation, l'accroissement du nombre d'activités porte la complexité de l'espace de recherche à la valeur indiquée par l'Equation 7.4.

$$Complexité(SLED - QRC1 \cup QRC2) = \sum_{i=1}^{nb_{act}} n_{Ai} = 3.6 \cdot 10^{46} \quad (7.4)$$

Le SMD obtenu va maintenant nous servir de point de départ pour identifier les schémas de contrôle à sélectionner au niveau de chacune des activités de la Mission, en faisant appel à l'algorithme de Bennour (Énergie/Durée) puis de Viterbi (Énergie) pour satisfaire l'ensemble de nos contraintes de performance y compris celles d'incertitude de Localisation.

TABLE 7.10 – PANORAMA(SLED - QRC1  $\cup$  QRC2) : Scénario de Mission Détaillé - Partie 1 - Aller

Objectif ( $X_{deb} - X_{fin}$ )	O <sub>I</sub> (0 – 31.5)						O <sub>II</sub> (31.5)		O <sub>III</sub> (31.5)	O <sub>IV</sub> (31.5)	O <sub>V</sub> (31.5 – 89.15)					
	DEP   LOC						ROT	LOC	ANIM	ROT	DEP   LOC					
Informations	X <sub>AI</sub>	0	6	9	14	25	25.5	31.5	31.5	31.5	35	55	57	58	60	
	dis <sub>AI</sub>	6	3	5	11	0.5	6	0	0	0	20	2	1	3	4.2	
	dur <sub>AI</sub>	-	-	-	-	-	-	10	1	10	-	-	-	-	-	
Contraintes Expérimentales	C <sub>1EXP</sub>															
	C <sub>2EXP</sub>															
	C <sub>3EXP</sub>															
	C <sub>4EXP</sub>															
	C <sub>5EXP</sub>															
Contraintes matérielle	C <sub>1MAT</sub>															
Contrainte de Localisation	C <sub>Loc</sub>	L1	L2	L2	L2	L2	L3	-	-	-	L2	L2	L4	L4	L5	
	1															
Méthode de localisation possible	2															
	3															
	4															
	5															
	EO															
Contraintes de sécurité	Inno cuité															
	Activité	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>	A <sub>10</sub>	A <sub>11</sub>	A <sub>12</sub>	A <sub>13</sub>	A <sub>14</sub>	A <sub>15</sub>
$\eta_{AI}$	6	10	15	25	15	15	15	5	5	15	25	20	15	10	15	

**Code couleur des activités**

Robot uniquement en rotation	Robot immobile	$V_{max} = 0.33$ m/s	$V_{max}$ fonction des capteurs utilisés

TABLE 7.11 – PANORAMA(SLED - QRC1 ∪ QRC2) : Scénario de Mission Détaillé - Partie 2 - Aller / Début retour

Objectif ( $x_{\text{deb}} - x_{\text{fin}}$ )	O <sub>v</sub> (31.5 – 89.15)										O <sub>VI</sub> (89.15)	O <sub>VII</sub> (89.15)	O <sub>VIII</sub> (89.15)	O <sub>V</sub> (89.15 – 170.3)
	DEP   LOC													
Tâches Robotiques	X <sub>AI</sub>	64.2	66	67	75	77	79	81	83	85.5	89.15	89.15	89.15	89.15
Informations	dis <sub>AI</sub>	1.8	1	8	2	2	2	2	2.5	3.65	0	0	0	6.15
	dur <sub>AI</sub>	-	-	-	-	-	-	-	-	-	10	1	10	-
Contraintes Expé- rimentales	C <sub>1EXP</sub>													
	C <sub>2EXP</sub>													
	C <sub>3EXP</sub>													
	C <sub>4EXP</sub>													
	C <sub>5EXP</sub>													
Contraintes matérielle	C <sub>1MAT</sub>													
Contrainte de Localisation	C <sub>Loc</sub>	L5	L5	L5	L5	L5	L5	L5	L6	L3	-	-	-	L6
Méthode de localisation possible	1													
	2													
	3													
	4													
	5													
Contraintes de sécurité	EO													
	Inno cuité													
Activité	A <sub>16</sub>	A <sub>16</sub>	A <sub>17</sub>	A <sub>18</sub>	A <sub>19</sub>	A <sub>20</sub>	A <sub>21</sub>	A <sub>22</sub>	A <sub>23</sub>	A <sub>24</sub>	A <sub>25</sub>	A <sub>26</sub>	A <sub>27</sub>	A <sub>28</sub>
n <sub>AI</sub>	9	15	15	15	25	20	15	10	6	6	3	1	3	6

**Code couleur des activités**

	Robot uniquement en rotation		Robot immobile		V <sub>max</sub> = 0.33 m/s		V <sub>max</sub> fonction des capteurs utilisés
--	------------------------------	--	----------------	--	-----------------------------	--	---

TABLE 7.12 – PANORAMA(SLED – QRC1 ∪ QRC2) : Scénario de Mission Détaillé – Partie 3 – Fin retour

Objectif ( $X_{deb} - X_{fin}$ )		Oix (89.15 – 170.3)																
		DEP   LOC																
Informations	$X_{AI}$	95.3	97.3	99.3	101.3	103.3	111.3	112.3	114.1	118.3	120.3	121.3	123.3	143.3	153.3	164.3	169.3	172.3
	$dis_{AI}$	2	2	2	2	8	1	1.8	4.2	2	1	2	20	10	11	5	3	6
Contraintes Expéri- mentales	$dur_{AI}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	$C_{1EXP}$																	
	$C_{2EXP}$																	
	$C_{3EXP}$																	
	$C_{4EXP}$																	
Contraintes matérielle	$C_{5EXP}$																	
	$C_{1Mat}$																	
Contrainte de Localisation	$C_{Loc}$	L5	L5	L5	L5	L5	L5	L5	L5	L4	L4	L2	L2	L2	L2	L2	L2	L1
	1																	
Méthode de localisation possible	2																	
	3																	
	4																	
	5																	
	EO																	
Contraintes de sécurité	Inno cuité																	
	Activité	A <sub>29</sub>	A <sub>30</sub>	A <sub>31</sub>	A <sub>32</sub>	A <sub>33</sub>	A <sub>34</sub>	A <sub>35</sub>	A <sub>36</sub>	A <sub>37</sub>	A <sub>38</sub>	A <sub>39</sub>	A <sub>40</sub>	A <sub>41</sub>	A <sub>42</sub>	A <sub>43</sub>	A <sub>44</sub>	A <sub>45</sub>
$n_{AI}$	10	15	20	25	15	15	9	15	10	15	15	20	25	15	25	15	10	6

**Code couleur des activités**

	Robot uniquement en rotation		Robot immobile		$V_{max} = 0.33$ m/s		$V_{max}$ fonction des capteurs utilisés
--	------------------------------	--	----------------	--	----------------------	--	--

### 7.3.3.2 PANORAMA(SLED2 - QRC1 $\cup$ QRC2) - Viterbi : Détermination de la Solution d'Allocation de Ressources

Dans cette partie nous allons détailler la construction de la Solution d'Allocation de Ressources SAR.

La prise en compte de la garantie de Localisation et de la dimension énergétique grâce à l'utilisation de l'algorithme de Viterbi dans PANORAMA(SLED2) conduit à la SAR (*Solution d'Allocation de Ressources*) présentée dans la Table 7.13. On y trouve l'indication des schémas de contrôle à utiliser ainsi que leur position de lancement prévue  $x_{deb}$ . La colonne Init matérialise les méthodes de Localisation qui doivent être initialisées avant leur mise en œuvre effective (fond non blanc).

Xdeb (m)	Schéma de contrôle	Init	Xdeb (m)	Schéma de contrôle	Init	Xdeb (m)	Schéma de contrôle	Init	Xdeb (m)	Schéma de contrôle	Init
0	(1/1)		56.89	(4/4)	2	85.5	(1/2)		115.53	(2/3)	2
4.4	(1/1)	2	57	(4/4)	2	88.15	(1/2)	2	118.3	(2/2)	
6	(1/2)		58	(2/2)		89.15	(6/1)	2	119.7	(2/2)	4
7.2	(1/2)	3	58.09	(2/2)	3	89.15	(7)	2	120.3	(4/2)	4
9	(2/3)		60	(2/3)		89.15	(6/2)		121.3	(4/4)	
12.4	(2/3)	4	64.2	(1/3)		89.15	(1/2)		123.3	(4/4)	
14	(4/4)		66	(1/3)		95.3	(1/2)		143.3	(5/4)	
25	(5/4)		67	(2/3)		97.3	(2/2)		153.3	(4/4)	
25.5	(5/4)		73.4	(2/3)	4	97.68	(2/2)	3	160.5	(4/4)	3
31.5	(6/1)		75	(4/4)		99.3	(2/3)		164.3	(2/3)	
31.5	(7)		77	(4/4)		101.3	(2/3)		167.6	(2/3)	2
31.5	(6/1)	4	79	(4/4)		103.3	(2/3)		169.3	(1/2)	
31.5	(5/4)		79.7	(4/4)	2	111.3	(1/3)		172.3	(1/2)	
35	(4/4)		81	(2/2)		112.3	(1/3)				
55	(4/4)		83	(1/2)		114.1	(2/3)				

TABLE 7.13 – PANORAMA(SLED2 - QRC1  $\cup$  QRC2) : Solution d'Allocation de Ressources

A partir de la Table 7.13 analysons dans un premier temps plus précisément le processus d'initialisation des méthodes de Localisation :

- Le fond bleu repère les initialisations se déroulant au niveau des vannes sans déplacement linéaire. On peut remarquer que l'initialisation de la Kinect commence avant l'arrivée à la vanne 2 ( $x = 88.15$ ) pour que KINODO soit opérationnelle dès la fin de l'analyse d'image.
- Le fond orange repère une activité pendant laquelle une initialisation est déployée mais dont la durée est supérieure ou égale à la durée de cette activité. Cela n'induit donc pas la création d'une nouvelle activité dans le SMD.
- Le fond vert indique les activités du SMD(SLED2) qui ont été décomposées. La durée d'initialisation étant inférieure à la durée de l'activité. On en dénombre 13.
- En conclusion le SMD(SLED2) final contient donc bien  $45 + 13 = 58$  activités (+ 29%) pour lesquelles les schémas de contrôle ont été sélectionnés par PANORAMA(SLED2).
- L'initialisation d'une méthode conduit parfois l'algorithme de Viterbi à définir des activités très courtes comme c'est le cas à  $X_S = 58m$  pour  $A_{14}$ .
- Lorsque les ultrasons sont nécessaires soit on utilise LZA pour se localiser et l'évitement d'obstacle est déployé grâce à SCH-LAS1-2-US, soit on emploie SCH-US car les lasers dans ces zones n'apportent aucun avantage de vitesse.

- Lorsque la méthode KINODO est utilisée on fait appel à SCH-LAS1 pour le déplacement et l'évitement d'obstacle de façon à bénéficier d'une vitesse supérieure de déplacement par rapport à SCH-US. Ce qui permet de fait de réduire l'énergie consommée pour parcourir l'activité concernée.

La Table 7.14 synthétise les principales informations sur les schémas utilisés tout au long de la mission. De façon à simplifier le décompte indiqué, tant en temps qu'en distance, les consommations d'initialisation des méthodes ont été intégrées aux indications énergétiques de la méthode courante.

- Contrairement à la proposition affichée dans SED, le schéma ODO seul nous permet d'assurer la localisation durant les 6 premiers mètres de notre mission malgré la dérive odométrique puisque la pose initiale du robot est supposée parfaitement connue.
- La méthode de localisation la plus utilisée est la méthode globale LZA-ODO. Elle représente plus de 50% de la longueur totale de la mission. On la retrouve dans toutes les activités dont le schéma de contrôle est en gras dans la Table 7.13.

Schéma de contrôle	DEP (Ressources matérielles)	LOC	Longueur utilisation (m)	% Longueur Mission	%Énergie
(1/1)	SCH-US (US)	ODO	6	3.36	2.18
(1/2)	SCH-US (US - Kinect)	KINODO	26.3	14.75	12.26
(1/3)	SCH-US (US LAS1-2 - Kinect)	COMPODOKIN	5.6	3.14	3.67
(2/2)	SCH-LAS1 (LAS1 - Kinect)	KINODO	8	4.49	4.28
(2/3)	SCH-LAS1 (LAS1 - Kinect)	COMPODOKIN	38.4	21.54	22.10
(4/2)	SCH-LAS1-2 (LAS1-2 - Kinect)	KINODO	1	0.56	0.58
(4/4)	SCH-LAS1-2 (LAS1-2)	LZA-ODO	73	40.94	35.26
(5/4)	SCH-LAS1-2-US (US LAS1-2)	LZA-ODO	20	11.22	10.66
(6/1)	ROT-ODO	ODO	0	0	4.70
(6/2)	ROT-ODO (LAS1-2 - Kinect)	KINODO	0	0	1.97
(7)	ANIM (Kinect)	/	0	0	2.34

TABLE 7.14 – PANORAMA(SLED2 - QRC1  $\cup$  QRC2) : Analyse de la Solution d'Allocation de Ressources

Passons maintenant à la présentation des résultats obtenus en simulation.

### 7.3.3.3 PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Simulation

Pour la SAR que nous venons de déterminer la simulation prévoit les marges de Temps et d'Énergie suivantes :

$$M_{TEMPES}(SLED2 - QRC1 \cup QRC2)_{th} = Objectif_{Durée} - Durée_{th} = 81.54 \text{ s} \quad (7.5)$$

$$M_{ENERGIE}(SLED2 - QRC1 \cup QRC2)_{th} = Objectif_{Energie} - Energie_{th} = 0.56 \text{ Wh} \quad (7.6)$$



On constate donc que ces 2 marges sont très confortables et que la marge énergétique SLED2 a été logiquement fortement accrue par rapport à celle affichée par la méthodologie PANORAMA(SED) grâce à l'utilisation de l'algorithme de Viterbi.

Passons maintenant à la dimension Localisation qui nous intéresse plus particulièrement. Les courbes de la Figure 7.2 présentent les incertitudes prévues en  $X_S$ ,  $Y_S$  et  $\theta$ .  $X_S$  et  $Y_S$  sont déterminés en fonction de l'orientation du robot. On y trouve en rouge la contrainte de performance imposée par l'utilisateur et en bleu l'incertitude calculée à partir de nos modèles et du SAR(SLED2). Seules les valeurs extrémales d'incertitudes sont reportées pour une activité et non leur évolution tout au long de parcours.

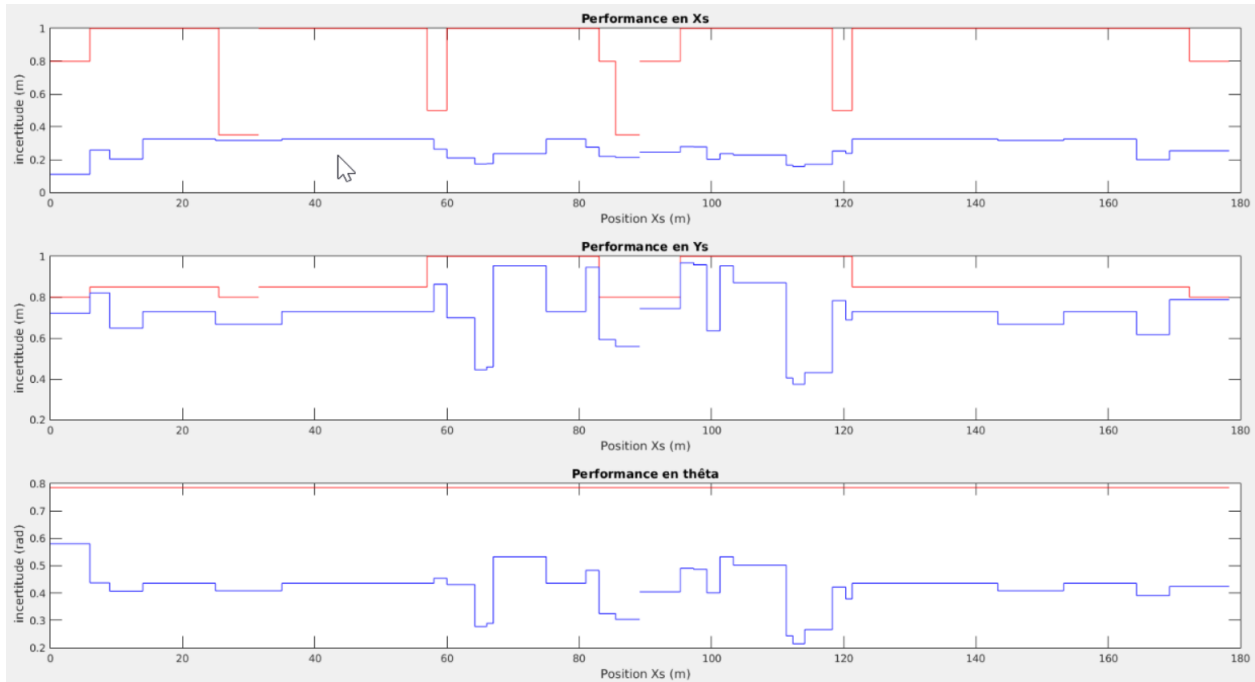


FIGURE 7.2 – PANORAMA(SLED2 - QRC1  $\cup$  QRC2) : Évolution simulée de la performance de Pose

L'analyse qualitative de ces courbes prévisionnelles montre que :

- Globalement la marge d'incertitude en  $X_S$  reste assez importante puisqu'elle n'est inférieure à 0.1m que pendant 3.4% de la mission.
- En revanche, selon  $Y_S$ , la marge d'incertitude est inférieure à 0.1m pendant 22.5% de la mission soit presque un quart de celle-ci.
- La marge d'incertitude en Thêta reste excellente sur toute la mission.

Les pavés d'incertitude suivants peuvent être établis pour les marges d'incertitudes de pose.

- Marge d'incertitude selon l'axe  $X_S$  :  $Mission([x]) = [0.032813, 0.84137]$  en m
- Marge d'incertitude selon l'axe  $Y_S$  :  $Mission([y]) = [0.011148, 0.62656]$  en m
- Marge d'incertitude d'orientation :  $Mission([\theta]) = [0.20534, 0.57121]$  en rad

Incertaince 13 : Pavés des marges d'incertitudes de la mission simulée  $QRC1 \cup QRC2$

L'analyse qualitative des marges d'incertitude de localisation calculées en simulation montre qu'elles sont parfois réduites. Pour que les expérimentations soient concluantes il faudra d'une

part que les modèles que nous avons développés soient valides sur l'ensemble de la mission et que, d'autre part, les hypothèses effectuées soient vérifiées. Entre autre, nous avons supposé que nous pouvions toujours détecter les QR-Codes verticaux entre 2m et 3m. Confrontons nous donc maintenant à l'expérimentation.

### 7.3.3.4 PANORAMA(SLED2 - QRC1 $\cup$ QRC2) : Analyse détaillée d'une mission

Nous avons réalisé plusieurs expériences mettant en œuvre le SAR(SLED2) que nous venons de décrire. Nous présentons dans ce paragraphe les relevés expérimentaux d'une mission représentative de l'ensemble de celles effectuées. Nous commencerons tout d'abord par présenter le profil de vitesse de la mission puis l'évolution des marges de temps et d'énergie, avant de nous attarder sur la variation des incertitudes de pose. Celles-ci sont mesurées de la façon suivante : à chaque recalage expérimental, on applique le modèle d'incertitude de la méthode de localisation en cours d'exécution. Puis entre 2 recalages nous jouons notre modèle de dérive odométrique.

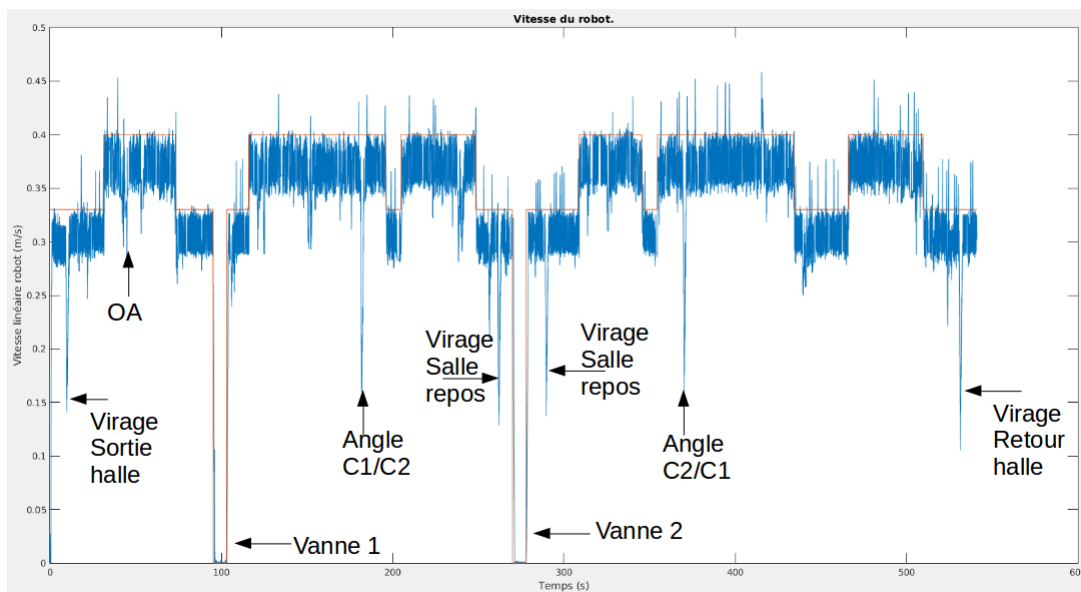


FIGURE 7.3 – PANORAMA(SLED2 - QRC1  $\cup$  QRC2) : Évolution mesurée de la vitesse du robot

La Figure 7.3 montre les profils de vitesse prévus et mesurés tout au long de la mission conformément au profil défini. La vitesse du robot dépend tout d'abord des contraintes de sécurité. Les vitesses demandées sont indiquées par un trait rouge continu. Dans les zones vitrées réelles ou simulées, et dans les zones où les lasers ne peuvent détecter tous les obstacles, les sonars sont indispensables et  $v_{Rob} = 0.33m/s$ . Dans les zones où la sécurité est assurée à l'aide de lasers,  $v_{Rob} = 0.4m/s$ . Il existe aussi une contrainte imposée pour obtenir des images exploitables. Ainsi lorsque la Kinect est utilisée on peut, au plus, se déplacer à 0.5 m/s pour ne pas flouter les images. Mais cette limitation est supplantée par les contraintes de sécurité. Bien évidemment la vitesse linéaire est nulle au niveau des vannes. On peut cependant constater que la vitesse maximale mesurée est légèrement inférieure à celle prévue. Ce phénomène peut rallonger notablement (10%) la durée des activités et donc l'énergie consommée entamant par la même les marges correspondantes. On rencontre aussi parfois des évitements d'obstacles pouvant être induits par des incertitudes de cap ou par des trajectoires trop proches d'obstacles non ou peu considérés dans la création du chemin. Dès lors on observe une variation importante de la vitesse linéaire pour contourner la zone encombrée.

La Figure 7.4 présente l'évolution mesurée de la marge de temps. On peut constater qu'elle alterne des phases de réduction avec des phases d'augmentation. Les premières sont induites par le fait que la durée d'une activité a été supérieure à celle prévue. Cela correspond à des zones où la vitesse du robot est légèrement inférieure à celle planifiée, ou à une trajectoire plus ou moins éloignée de celle prévue (ligne droite). Au contraire, les secondes qui accroissent la marge de temps, sont induites par des activités dont la durée a été moindre que celle prévue. C'est entre autre le cas au niveau des vannes 1 et 2 ( $t=100s$  et  $280s$ ) puisque nous prévoyons des durées de rotation et d'analyse d'image bien supérieures à celles observées pratiquement. Quoiqu'il en soit la marge de temps finale de plus d'une minute (65.4 s) reste très confortable même si elle a été réduite de 19% pour une mission rallongée de seulement 3% de la durée prévue.

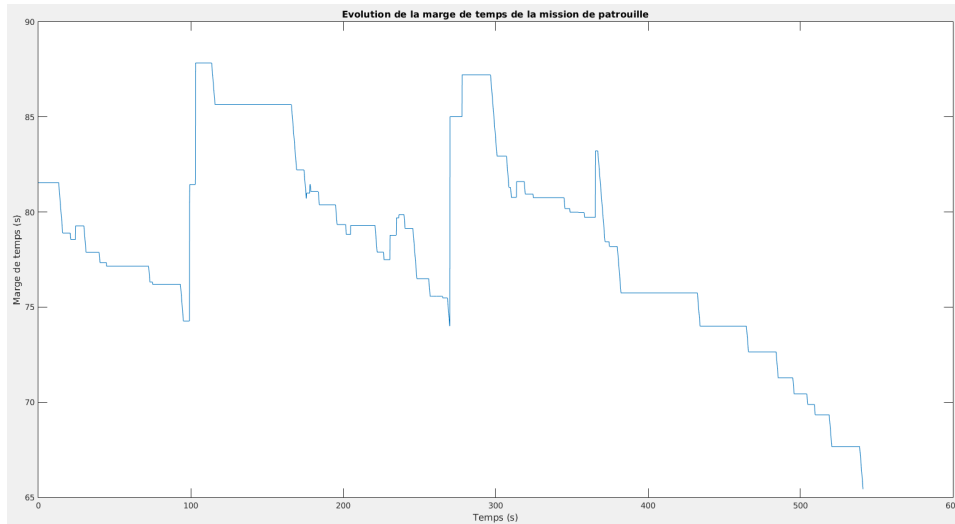


FIGURE 7.4 – PANORAMA(SLED2 - QRC1  $\cup$  QRC2) : Évolution de la marge de temps

La Figure 7.5 présente l'évolution mesurée de la marge d'énergie. Là aussi on observe une alternance de phases de décroissance et de croissance. Les premières sont à corréliser avec celles où la marge de temps diminue. Si l'on reste plus longtemps que prévu dans une activité on consommera plus d'énergie que planifié. Mais il ne faut pas aussi oublier que notre modèle énergétique de déplacement, bien que d'assez bonne qualité (erreur de 10% [112]) reste inexact. Les phases de croissance sont rencontrées pour les mêmes raisons que précédemment au niveau des vannes 1 et 2. La marge d'énergie finale reste acceptable à plus de 0.4 Wh soit 28% de la marge initiale, ce qui correspond de fait à une consommation de 20% supérieure à celle prévue. On peut s'étonner qu'un rallongement de seulement 3% de la durée de la mission induise un surcoût énergétique de 20%. Cela confirme que notre modèle énergétique reste encore perfectible.

Passons maintenant à l'analyse de l'évolution mesurée de l'incertitude de pose du robot tout au long de la mission présentée dans la Figure 7.6

Commençons par nous attarder sur les formes observées des incertitudes mesurées en nous focalisant sur l'évolution le long de  $Y_S$  :

- En tout début de mission l'incertitude croît conformément à la dérive odométrique.
- L'utilisation de LZA-ODO se caractérise par une alternance de localisation et de dérive odométrique d'amplitude réduite qui est observée à l'aller avant et après la vanne 1 dans le couloir 1 ( $44.8 s < t < 178.3 s$ ) puis sur une grande partie du retour à nouveau dans le couloir 1 ( $375.2 s < t < 496.7 s$ ). On constate que LZA-ODO semble jouer parfaitement son rôle en bornant fortement l'incertitude de localisation.

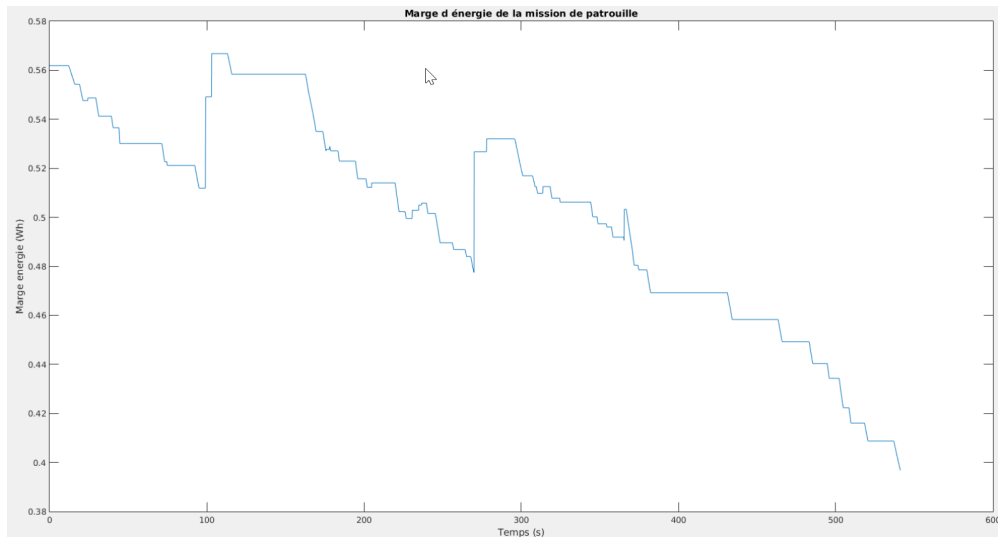


FIGURE 7.5 – PANORAMA(SLED2 - QRC1  $\cup$  QRC2) : Évolution de la marge d'énergie

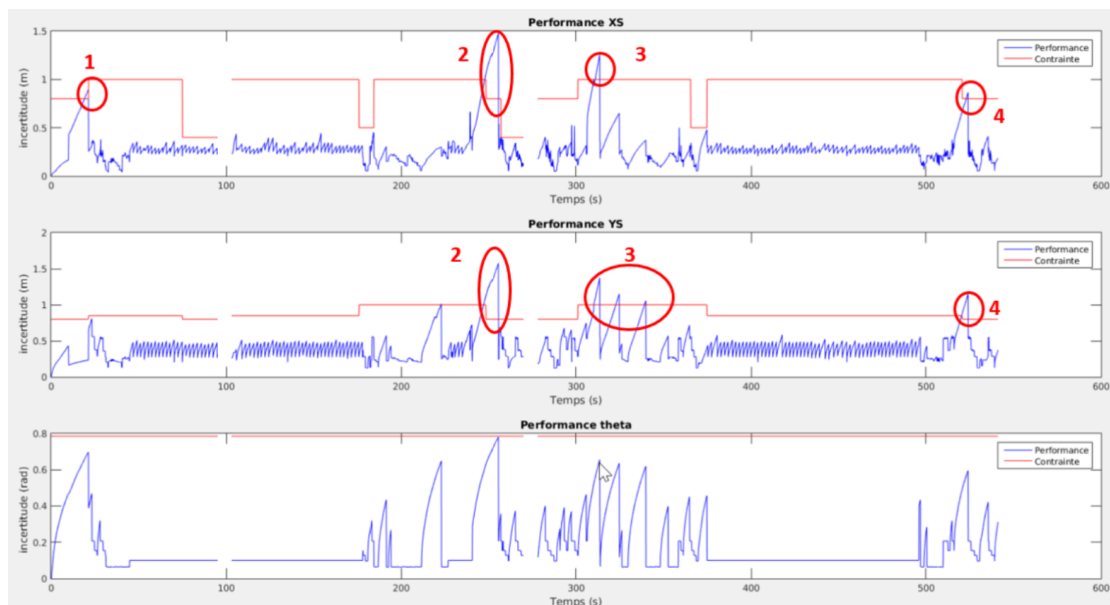


FIGURE 7.6 – PANORAMA(SLED2) (QRC1  $\cup$  QRC2) : Évolution mesurée de l'incertitude de Pose

- Le reste du temps, sauf en tout début de mission, ce sont les méthodes KINODO et COM-PODOKIN qui sont employées. L'évolution se caractérise alors comme un accroissement de l'incertitude de localisation en raison de la dérive odométrique jusqu'au moment où la Kinect détecte enfin un QR-Code. L'incertitude décroît alors en paliers successifs au fur et à mesure que le robot se rapproche du marqueur. La différence entre les 2 méthodes ne s'observe qu'à travers de l'analyse de l'incertitude en Thêta qui est plus réduite pour COMPODOKIN que pour KINODO.

Si l'on effectue maintenant une analyse critique de l'évolution des incertitudes mesurées on peut énoncer que :

- Tout d'abord on peut tout de même constater que les contraintes de localisation imposées sont satisfaites sur presque toute la mission. Pleinement pour l'incertitude en Thêta. Sur environ 95% de la mission pour les incertitudes selon  $X_S$  et  $Y_S$ . On peut donc malheureusement conclure que la garantie d'incertitude de pose recherchée n'est donc pas atteinte.

- Les violations de contraintes de localisation en  $X_S$  et  $Y_S$  se situent dans les mêmes zones de la mission même si elles sont plus prononcées selon  $Y_S$ .
- Les zones de violation sont repérées par les numéros suivants dans la figure : (1) la sortie de la salle d'expérimentation à l'aller, (2)(3) la zone de repos au voisinage de la vanne 2 à l'aller et au retour, (4) l'arrivée dans la salle d'expérimentation au retour.
- L'origine de ces violations parfois minimales (comme en (1)) est toujours la même. Le robot fait appel à la méthode KINODO ou COMPODOKIN pour se localiser et est en train de se déplacer à l'odométrie dans l'attente de détecter un QR-Code pour se recalibrer. Malheureusement ce dernier est détecté trop tardivement et cela a engendré une violation de la contrainte de localisation imposée en raison de l'augmentation de la dérive odométrique.
- Comme nous l'avons pressenti, l'incertitude de cap et l'angle d'ouverture de la Kinect 1 n'apportent pas de garantie de détection des QR-Codes verticaux malgré une instrumentation de l'environnement qui semble suffisante. C'est la non détection de certains marqueurs qui allonge la durée de dérive odométrique et conduit à violer les contraintes de localisation imposées.

### 7.3.3.5 Conclusion

La méthodologie PANORAMA(SLED2) a permis de construire un Scénario de Mission Détaillé et de trouver une Solution d'Allocation de Ressources avec un environnement équipé de marqueurs verticaux. La simulation a montré que les marges d'incertitude de localisation escomptées étaient parfois faibles surtout selon  $Y_S$ . Les contraintes de performances de Sécurité, Durée et Énergie ont été satisfaites expérimentalement. En revanche les craintes que nous avons formulées quant à la capacité à réaliser expérimentalement la mission fixée avec les limites d'incertitudes imposées ont malheureusement été vérifiées. Les garanties d'incertitude de Localisation demandées n'ont donc pu être apportées. Pour palier ce problème, nous allons doter l'environnement de nouveaux marqueurs pouvant nous apporter une garantie de détection dans les zones problématiques identifiées.

### 7.3.4 Conclusion

Il est tout de même extrêmement frustrant de devoir sur-instrumenter l'environnement pour pouvoir effectuer la mission proposée avec des performances de localisation somme toute assez modestes mais garanties. Ce constat n'est absolument pas dû à des limitations de l'approche proposée. Il trouve essentiellement son origine dans 2 raisons très différentes. En premier lieu l'approche de garantie est nécessairement conservatrice et parfois, nos modèles le sont trop, ce qui peut conduire à surestimer les incertitudes pouvant être atteintes. Mais surtout elle est évidemment dépendante des méthodes utilisées et des performances des capteurs employés. Nous avons montré dans ce manuscrit que nous ne pouvons garantir de pouvoir toujours détecter les marqueurs verticaux en raison de leur angle d'ouverture et des erreurs de cap qui peuvent être rencontrées. Pour surmonter cette difficulté nous allons faire appel aux marqueurs horizontaux que nous avons placés au plafond aux endroits critiques de la mission.

## 7.4 Utilisation des QR-Codes verticaux et horizontaux

Dans cette partie l'ensemble QRC3 des marqueurs horizontaux va venir remplacer l'ensemble QRC2 des marqueurs verticaux que nous avons ajoutés dans la partie précédente. Nous travaillerons donc avec l'ensemble de QR-Codes  $QRC1 \cup QRC3$ .

L'utilisation de la Kinect 2 très énergivore par rapport aux autres capteurs (Annexe A) nous conduit à réévaluer notre contrainte énergétique à la hausse pour faire face au surcoût qu'engendrera son utilisation. Nous estimons cette surconsommation à 0.6 Wh environ en supposant que la Kinect 2 pourrait être utilisée, dans le pire cas, sur la moitié de la mission, ce qui correspond environ à la zone qui a été équipée de marqueurs au plafond. Ainsi la contrainte énergétique de fin de mission est donnée par la Table 7.15.

Axe de Performance	Localisation (m) ( $x_{deb} - x_{fin}$ ) ou $x_{loc}$	Valeurs (m, m, rad) ( $X_s, Y_s, \theta$ )	Description
Énergie	178.3	Énergie robot $\leq 3$ Wh	Énergie maximale que le robot peut consommer

TABLE 7.15 – Performance énergétique à garantir avec QRC1  $\cup$  QRC3

#### 7.4.1 PANORAMA(SED - QRC1 $\cup$ QRC3) : Scénario de Mission Détaillé

Pour le calcul du scénario de mission détaillé, l'ajout de l'utilisation de la Kinect 2 ne change rien au découpage de la mission SED présenté dans les paragraphes précédents car elle est supposée utilisable partout. La méthode KIN2ODO est toujours utilisable, mais, dans les zones où il n'existe pas de QR-Codes au plafond, le déplacement se fera à l'odométrie.

Nous ne détaillerons donc ici que la Solution d'Allocation de Ressources correspondant au SMD décomposant la mission en 37 Activités comme précédemment. Cependant la complexité associée est plus importante car l'introduction de KIN2ODO accroît le nombre de schémas éligibles dans la majorité des activités

$$Complexité(SED - QRC1 \cup QRC3) = \sum_{i=1}^{nb_{act}} n_{Ai} = 2.43 \cdot 10^{42} \quad (7.7)$$

Xdeb (m)	Schéma de contrôle	Xdeb (m)	Schéma de contrôle	Xdeb (m)	Schéma de contrôle	Xdeb (m)	Schéma de contrôle	Xdeb (m)	Schéma de contrôle
0	(5/6)	31.5	(5/6)	73	(5/6)	99.3	(5/6)	143.3	(1/1)
6	(5/6)	35	(5/6)	79	(5/6)	103.3	(5/6)	153.3	(1/1)
9	(5/6)	57	(5/6)	83	(5/6)	109.3	(5/6)	166.3	(1/1)
12	(5/6)	60	(5/6)	89.15	(6/6)	111.3	(5/6)	169.3	(1/1)
25	(5/6)	64	(5/6)	89.15	(7)	114.1	(5/6)	172.3	(1/1)
31.5	(6/6)	64.2	(5/6)	89.15	(6/6)	114.3	(5/6)		
31.5	(7)	67	(5/6)	89.15	(5/6)	118.3	(5/6)		
31.5	(6/6)	69	(5/6)	95.3	(5/6)	121.3	(1/1)		

TABLE 7.16 – PANORAMA(SED - QRC1  $\cup$  QRC3) : Scénario de Mission Détaillé et Solution d'Allocation de Ressources

On obtient le Scénario de Mission Détaillé de la Table 7.16 qui fait très largement appel à la Kinect 2 puisque dans PANORAMA(SED) on cherche à maximiser la consommation énergétique. L'algorithme de Bennour force le SAR à utiliser l'odométrie en fin de mission à partir

de  $x=121.3$  m. Cela est totalement cohérent avec l'approche PANORAMA(SED) qui suppose que la localisation peut-être assurée par les méthodes supposées valides dans une zone donnée.

De toute évidence, ces choix montrent bien que les contraintes de localisation qui doivent être satisfaites ne pourraient pas l'être avec ce scénario.

Les marges de temps et énergie données par les Équations 7.8 et 7.9 sont théoriquement disponibles. On peut constater que la marge de Temps a été fortement accrue. Cela provient du fait que SCH-LAS1-2-US/KIN2ODO consomme plus que SCH-US/COMPODOKIN nous permettant d'aller plus vite en sécurité tout en maximisant la consommation énergétique.

$$M_{TEMPS}(SED - QRC1 \cup QRC3)_{th} = Objectif_{Durée} - Durée_{th} = 67.22 \text{ s} \quad (7.8)$$

$$M_{ENERGIE}(SED - QRC1 \cup QRC3)_{th} = Objectif_{Energie} - Energie_{th} = 0.023 \text{ Wh} \quad (7.9)$$

Nous allons maintenant prendre explicitement en compte les contraintes de pose définies en utilisant PANORAMA(SLED2).

## 7.4.2 PANORAMA(SLED2 - QRC1 $\cup$ QRC3)

Nous allons dans un premier temps détailler la Solution d'Allocation de Ressources obtenue avec PANORAMA(SLED2) avant de présenter les résultats de la simulation. Dans un second temps nous détaillerons les mesures effectuées lors d'une de nos missions et nous finirons par une analyse plus globale de l'ensemble des expérimentations déployées.

### 7.4.2.1 PANORAMA(SLED2 - QRC1 $\cup$ QRC3) - Viterbi : Détermination de la Solution d'Allocation de Ressources

Tout comme PANORAMA(SED), le découpage initial du Scénario de Mission Détaillé est analogue à celui qui a été proposé plus haut. La mission est donc initialement composée de 45 activités. La complexité de l'espace d'état s'est à nouveau accrue avec l'ajout de KIN2ODO pour atteindre la valeur suivante :

$$Complexité(SLED2 - QRC1 \cup QRC3) = \sum_{i=1}^{nb_{act}} n_{Ai} = 2.46 \cdot 10^{51} \quad (7.10)$$

La solution d'Allocation de Ressources est détaillée dans la Table 7.17. On peut effectuer les commentaires suivants :

- Le fond bleu repère les initialisations se déroulant au niveau des vannes sans déplacement linéaire. On peut encore remarquer que l'initialisation de la Kinect 2 commence avant l'arrivée à la vanne 2 ( $x = 88.4$ ) pour que KIN2ODO soit opérationnelle dès la fin de l'analyse d'image. Le même phénomène est observé pour l'initialisation de LZA à la sortie de la vanne 1.
- Le fond orange repère une activité pendant laquelle une initialisation est déployée mais dont la durée est supérieure ou égale à la durée de cette activité. Cela n'induit donc pas la création d'une nouvelle activité dans le SMD.
- Le fond rouge signale que plusieurs méthodes de localisation sont initialisées simultanément. Cela est observé 3 fois.
- Le fond vert indique les activités du SMD(SLED2) qui ont été décomposées. La durée d'initialisation étant inférieure à la durée de l'activité. On en dénombre 13.

- En conclusion le SMD(SLED2) final contient  $45 + 13 = 58$  activités (+ 29%) pour lesquelles les schémas de contrôle ont été sélectionnés par PANORAMA(SLED2).
- L'initialisation d'une méthode conduit parfois l'algorithme de Viterbi à définir des activités très courtes. L'une d'elles n'a que 27 cm de long pour  $A_{44}$ .
- A  $X_S = 88.4m$  on initialise KIN2ODO alors que l'on est en train de l'utiliser. Cela semble paradoxal. Cela provient du fait qu'il n'y a pas de méthode de localisation associée à l'analyse d'image et que donc l'algorithme doit préparer la méthode de localisation qui devra être utilisée après la vanne.
- Globalement on peut constater que la prise en compte explicite de la localisation permet d'utiliser de façon pertinente la méthode KIN2ODO puisque, bien que la Kinect 2 soit utilisable dans la totalité de l'environnement, la position des amers du plafond est connue, ce qui permet d'estimer la dérive odométrique.

Xdeb (m)	Schéma de contrôle	Init	Xdeb (m)	Schéma de contrôle	Init	Xdeb (m)	Schéma de contrôle	Init	Xdeb (m)	Schéma de contrôle	Init
0	(1/1)		57.28	(4/4)	6	89.15	(6/1)	6	112.3	(1/3)	2
5.57	(1/1)	6	58	(2/6)		89.15	(7)	6	114.1	(2/3)	2
6	(2/6)		60	(2/6)		89.15	(6/6)		118.3	(2/2)	
9	(2/6)		64.2	(1/6)		89.15	(1/6)		118.7	(2/2)	4
12.4	(2/6)	4	66	(1/6)		95.3	(2/6)		120.3	(4/4)	
14	(4/4)		67	(2/6)		95.7	(2/6)	4	121.3	(4/4)	
25	(5/4)		73.4	(2/6)	4	97.3	(4/4)		123.3	(4/4)	
25.5	(5/4)		75	(4/4)		99.3	(4/4)		143.3	(5/4)	
31.5	(6/1)		77	(4/4)		99.55	(4/4)	6	153.3	(4/4)	
31.5	(7)		77.95	(4/4)	6	101.3	(2/6)		163	(4/4)	6
31.5	(6/1)	4	79	(2/6)		103.3	(2/6)		164.3	(2/6)	
31.5	(5/4)		81	(2/6)		109.7	(2/6)	4	169.3	(2/6)	
35	(4/4)		83	(1/6)		110.03	(2/6)	4/3	172.3	(1/6)	
55	(4/4)		85.5	(1/6)		111.03	(2/6)	4/3/2			
57	(4/4)		88.4	(1/6)	6	111.3	(5/4)	3-2			

TABLE 7.17 – PANORAMA(SLED2 - QRC1  $\cup$  QRC3 ) : Scénario de Mission Détaillé et Solution d'Allocation de Ressources

La Table 7.18 synthétise les principales informations sur les schémas utilisés tout au long de la mission. De façon à simplifier le décompte indiqué, tant en temps qu'en distance, les consommations d'initialisation des méthodes ont été intégrées aux indications énergétiques de la méthode courante.

- On dénombre l'utilisation de 11 schémas différents.
- Comme pour SLED2(QRC1  $\cup$  QRC2) la mission commence avec une localisation odométrique.
- Certains schémas sont utilisés sur une distance très courte comme SCH-US/COMPOKIN qui est utilisé sur 1.8m.
- La méthode de localisation la plus utilisée est toujours la méthode globale LZA-ODO. Elle représente encore plus de 50% de la longueur totale de la mission.
- La méthode de localisation KIN2ODO est employée sur 38% de la mission.



Schéma de contrôle	DEP (Ressources matérielles)	LOC	Longueur utilisation (m)	% Longueur Mission	%Énergie
(1/1)	SCH-US (US)	ODO	6	3.27	1.77
(1/3)	SCH-US (US LAS1-2 - Kinect)	COMPODOKIN	1.8	1	0.97
(1/6)	SCH-US (US – Kinect 2)	KIN2ODO	21	11.83	14.72
(2/2)	SCH-LAS1 (LAS1 - Kinect)	KINODO	2	1.12	0.92
(2/3)	SCH-LAS1 (LAS1-2 - Kinect)	COMPODOKIN	4.2	2.36	1.98
(2/6)	SCH-LAS1 (LAS1 – Kinect 2)	KIN2ODO	46.2	25.91	31.40
(4/4)	SCH-LAS1-2 (LAS1-2)	LZA-ODO	76	42.62	29.85
(5/4)	SCH-LAS1-2-US (US LAS1-2)	LZA-ODO	21	11.78	8.86
(6/1)	ROT-ODO	ODO	0	0	4.88
(6/6)	ROT-ODO (Kinect 2)	KIN2ODO	0	0	2.69
(7)	ANIM (Kinect)	/	0	0	1.96

TABLE 7.18 – PANORAMA(SLED2 - QRC1  $\cup$  QRC3 ) : Analyse de la Solution d’Allocation de Ressources

Abordons maintenant les résultats obtenus en simulation.

#### 7.4.2.2 PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Simulation

Commençons par présenter les marges de temps (Eq. 7.11) et énergétique (Eq. 7.12) prévisionnelles pour la SAR calculée. On peut constater que ces marges sont du même ordre de grandeur que celles obtenues par PANORAMA(SLED2 - QRC1  $\cup$  QRC2). Du point de vue énergétique cela confirme que la modification imposée sur la contrainte d’énergie était correctement estimée.

$$M_{TEMPS}(SLED2 - QRC1 \cup QRC3)_{th} = Objectif_{Durée} - Durée_{th} = 85.8 \text{ s} \quad (7.11)$$

$$M_{ENERGIE}(SLED2QRC1 \cup QRC3)_{th} = Objectif_{Energie} - Energie_{th} = 0.513 \text{ Wh} \quad (7.12)$$

En ce qui concerne l’évolution de l’incertitude de pose la Figure 7.7 peut être tracée. Comme pour PANORAMA(SED) on n’observe aucune violation de performance d’incertitude de localisation sur l’ensemble de la mission mais les marges minimales sont nettement supérieures. De plus l’utilisation de la méthode KINODO devrait nous garantir la détection des QR-Codes du plafond et donc limiter notre dérive odométrique.

Les pavés d’incertitude suivants peuvent être établis pour les marges d’incertitudes de pose de PANORAMA(SLED2 - QRC1  $\cup$  QRC3) :

- Marge d’incertitude selon l’axe  $X_S$  :  $Mission([x]) = [0.0828, 0.7888]$  en m
- Marge d’incertitude selon l’axe  $Y_S$  :  $Mission([y]) = [0.0610, 0.06412]$  en m
- Marge d’incertitude d’orientation :  $Mission([\theta]) = [0.2053, 1.1628]$  en rad

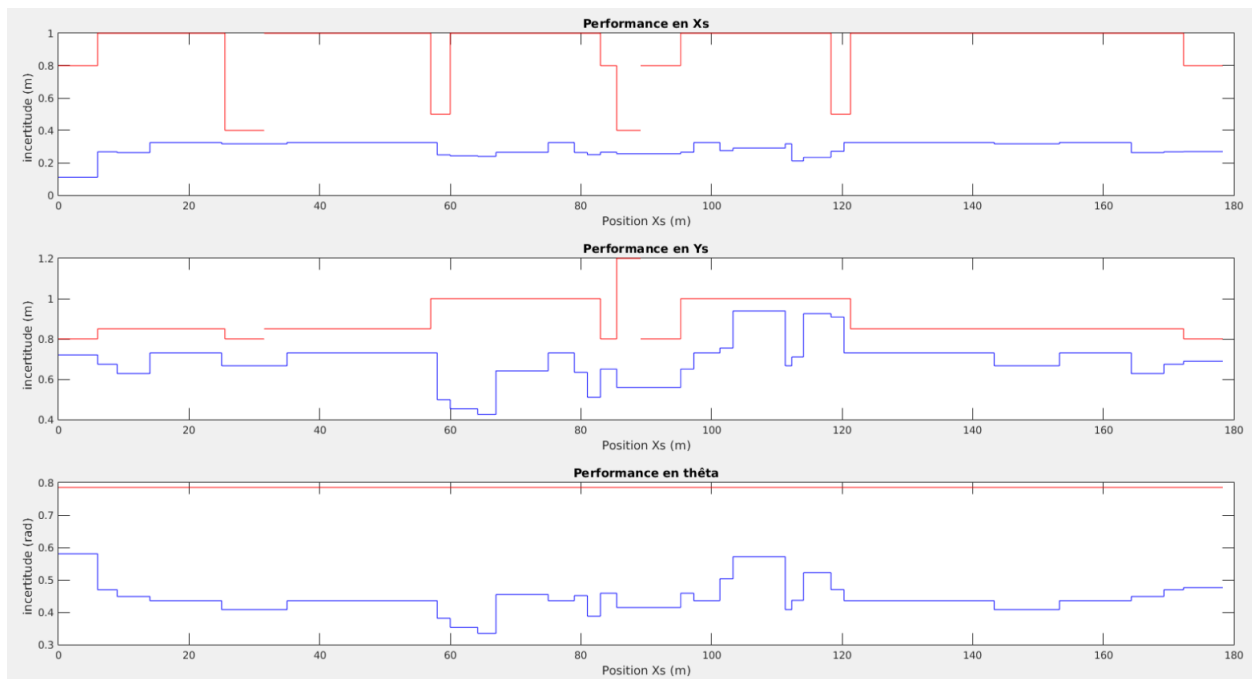


FIGURE 7.7 – PANORAMA(SLED2 - QRC1  $\cup$  QRC3) : Évolution simulée de la performance de Pose

Incertitude 14 : Pavés des marges d’incertitudes de la mission simulée  $QRC1 \cup QRC3$

Comparons maintenant ces résultats obtenus en simulation avec ceux d’une mission réalisée expérimentalement.

### 7.4.2.3 PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Analyse détaillée d’une mission

Nous avons effectué une quinzaine de mission de Patrouille. Nous proposons ici une analyse détaillée de l’une d’elles.

Commençons par présenter le profil de vitesses relevé. On peut constater comme précédemment que celles-ci évoluent entre  $v_{Rob} = 0.33m/s$  et  $v_{Rob} = 0.4m/s$  en fonction des contraintes sécuritaires et des capteurs utilisables. Les mêmes variations de vitesses sont aussi observées avec les mêmes origines (évitement d’obstacle, asservissement de la trajectoire).

L’évolution expérimentale de la marge de temps est présentée dans la Figure 7.9. Sa valeur finale de 69.6s est comparable à celle obtenue pour la mission réalisée avec PANORAMA(SLED2 - QRC1  $\cup$  QRC2). Les variations observées ont elles aussi les mêmes origines (durées d’activités plus ou moins respectées).

L’évolution de la marge d’énergie est représentée dans la Figure 7.10. Sa valeur finale de 0.26 Wh est légèrement inférieure à celle obtenue dans PANORAMA((SLED2 - QRC1  $\cup$  QRC2), mais reste confortable. Elle démontre que la mission n’était pas réalisable en faisant appel à la Kinect 2 avec la contrainte énergétique initiale imposée de 2.4 Wh puisque l’on a consommé environ 2.7 Wh pour réaliser la mission. Les variations observées ont elles aussi les mêmes origines (durées d’activités plus ou moins respectées).

Passons maintenant à l’analyse de l’évolution mesurée de l’incertitude de pose du robot tout au long de la mission présentée dans la Figure 7.11. On constate que conformément à notre attente, le recours à la méthode KIN2ODO permet de satisfaire les garanties d’incertitude de pose attendues. L’utilisation de la Kinect 2 et des marqueurs au plafond permet donc bien de garantir leur détection et donc les incertitudes envisagées.

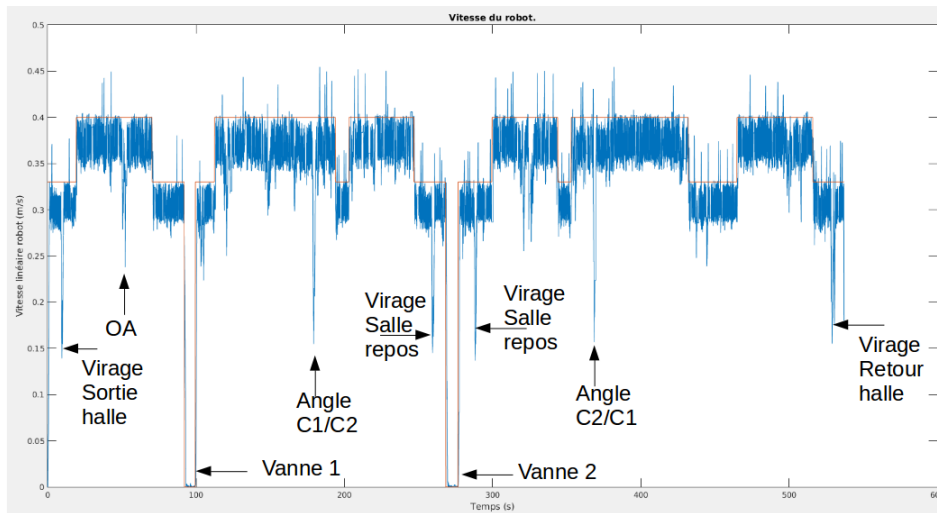


FIGURE 7.8 – PANORAMA(SLED2 - QRC1  $\cup$  QRC3) : Évolution mesurée de la vitesse du robot

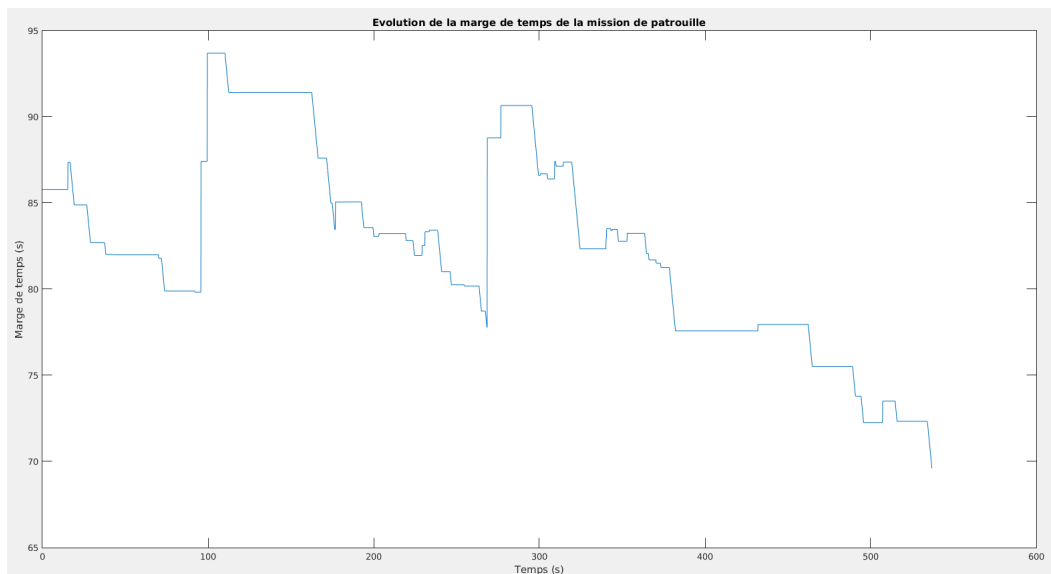


FIGURE 7.9 – PANORAMA(SLED2 - QRC1  $\cup$  QRC3) : Évolution de la marge de temps

Pour terminer, il est évidemment intéressant de comparer l'incertitude de pose que nous avons mesurée avec une réalité terrain. Celle-ci est difficile à obtenir dans un environnement assez étendu sans dispositif de mesure externe. Nous l'obtenons de la façon suivante. Durant toute l'expérimentation les données de la Kinect 1 et du compas sont enregistrées. Ces données sont alors utilisées hors-ligne pour reconstruire la réalité terrain en utilisant, les modèles d'incertitude de la méthodes COMPODOKIN. Pour que cette reconstruction soit possible il faut d'une part que la méthode COMPODOKIN ne soit pas utilisée par le robot pour se localiser. D'autre part il faut que les données marqueurs et compas soient disponibles simultanément ce qui n'est pas toujours le cas. Nous ne pourrons donc pas évaluer cette réalité terrain sur l'ensemble de la mission.

La Figure 7.12 présente le relevé des incertitudes de pose de notre mission (en bleu) et la réalité terrain lorsqu'elle peut être reconstruite (en rouge). On peut constater d'une part que, la plupart du temps nos incertitudes sont plus faibles que celles de la réalité terrain. D'autre part que, dans pratiquement toute la mission, il existe une intersection entre nos incertitudes

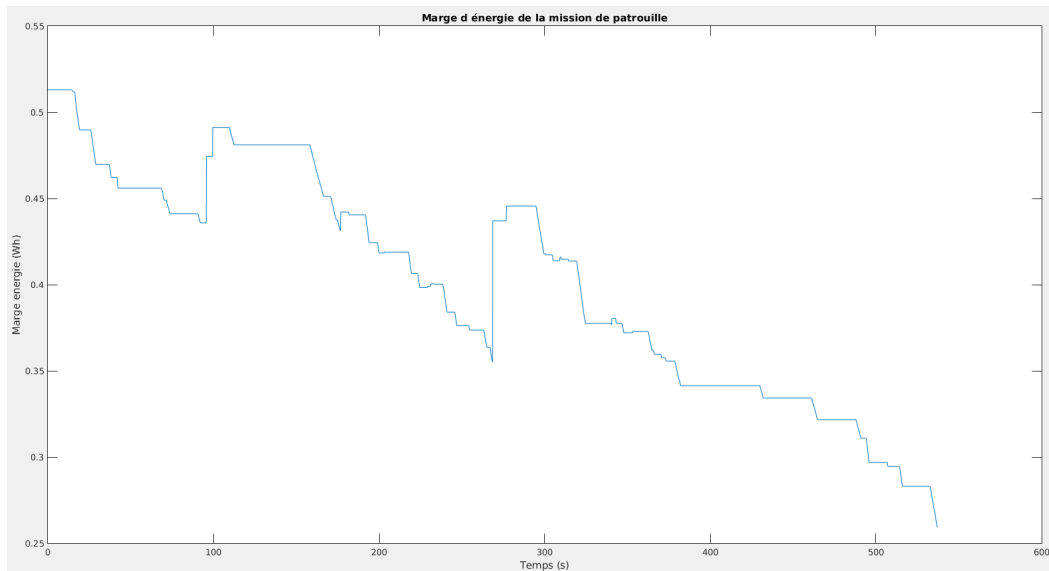


FIGURE 7.10 – PANORAMA(SLED2 - QRC1  $\cup$  QRC3) : Évolution de la marge d'énergie

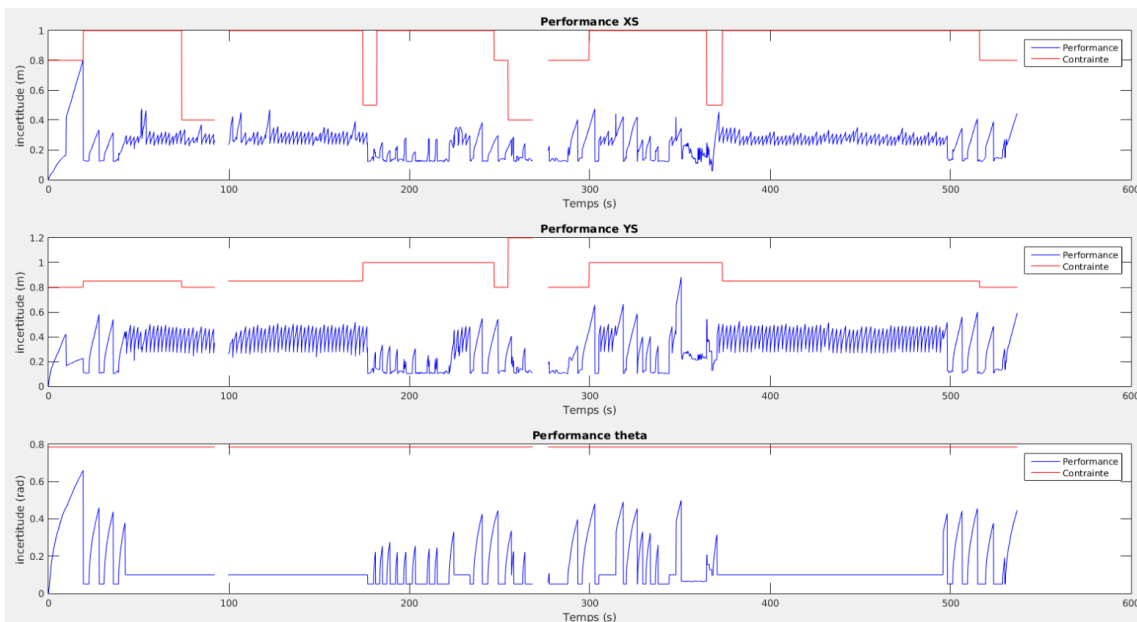


FIGURE 7.11 – PANORAMA(SLED2) (QRC1  $\cup$  QRC3) : Évolution mesurée de l'incertitude de pose

et la réalité terrain retenue ce qui revient à supposer que le robot se trouvait à l'intersection des 2 relevés. Enfin, on peut noter que, très rarement, il n'existe pas de recouvrement entre les incertitudes et la réalité terrain ce qui pointe une erreur d'estimation de la part de l'une des 2 sources d'information. Par exemple en  $X_S$  pour  $t = 43s$  l'origine du phénomène peut être due à une erreur momentanée de localisation de la méthode LZA rapidement corrigée. D'autre part les erreurs de cap parfois observées restent bien inférieures à la contrainte imposée.

L'analyse détaillée d'une mission vient d'être effectuée. Elle montre qu'il est possible de garantir l'exécution d'une mission tout en satisfaisant des contraintes de durée, de sécurité, d'énergie et d'incertitude de pose en faisant appel à une allocation éclairée des ressources matérielles et logicielles tout au long de la mission.

Nous allons terminer notre étude expérimentale en analysant globalement l'ensemble des missions exécutées.

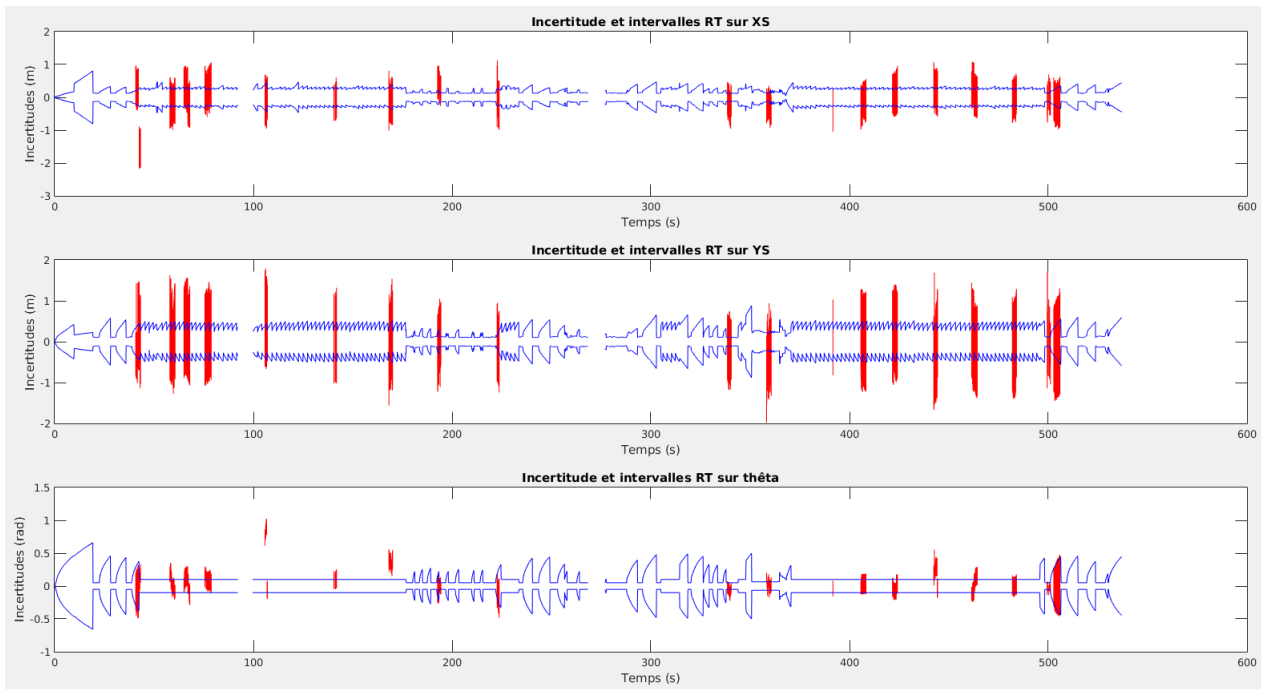


FIGURE 7.12 – PANORAMA(SLED2) (QRC1  $\cup$  QRC3) : Incertitude de pose mesurée et réalité terrain

#### 7.4.2.4 PANORAMA(SLED2 - QRC1 $\cup$ QRC3) : Analyse globale des missions réalisées

Au total nous avons réalisé 15 missions de Patrouille. Dans cette partie nous allons analyser globalement les résultats obtenus.

La Figure 7.13 décompose les missions exécutées en 4 groupes.

- 8 d’entre-elles, soit 54% ont été réalisées en respectant les contraintes d’incertitude de pose fixées et en obtenant des profils d’incertitudes similaires à ceux que nous avons analysés précédemment.
- 3 d’entre-elles, soit 20% se sont aussi terminées avec succès. Cependant si les marges de temps et d’énergie ont été satisfaites, les contraintes d’incertitude pose ont été parfois violées localement durant la mission. La méthode LZA-ODO a parfois eu un problème de localisation mais qui s’est solutionné au bout de quelques mètres. Ainsi le robot a tout de même pu achever sa mission.
- 2 d’entre-elles, soit 13% n’ont pu être menées à leur terme en raison de problème de localisation se prolongeant trop longtemps.
- 2 d’entre-elles, soit 13% n’ont pu s’achever en raison de problème matériel ou logiciel inhérent aux expérimentations. L’un provenait d’une déconnexion matérielle d’un capteur, l’autre d’un problème logiciel de communication avec un capteur.

Si l’on exclut les problèmes matériels ou logiciels on constate donc que malgré l’emploi de marqueurs horizontaux qui sont sûrs d’être détectés, il persiste parfois des problèmes de localisation. Cette perte de garantie peut, selon nous, avoir 2 origines.

- De façon globale notre approche de décomposition en activités n’intègre pas actuellement les incertitudes de localisation dans le processus de segmentation. Même si la plupart du temps le continuum fonctionnel des méthodes de localisation permet d’assurer la qualité du suivi de pose, il se peut fort bien qu’une méthode soit initialisée ou utilisée dans une

zone où elle est totalement inadaptée. Cela peut donc induire des ruptures dans la garantie de localisation.

- De façon plus spécifique et sans être exclusif de la remarque précédente, bien qu'étant globalement assez robuste, des ruptures de garanties sont parfois observées pour la méthode LZA. N'oublions pas que, pour cette méthode, notre modèle prédictif n'a été établi qu'à partir de données capteurs issues de notre seule zone d'expérimentation du couloir 2. Il n'est donc pas exclu que cela en limite quelque peu la validité et induise des dysfonctionnements locaux surtout dans cet environnement à fort aliasing perceptuel. L'amélioration de la robustesse de LZA devrait permettre d'accroître encore significativement le taux de garantie assuré.

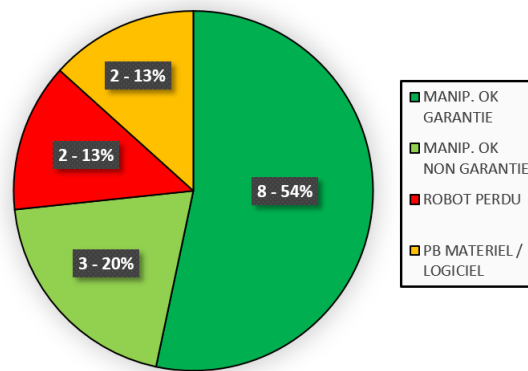


FIGURE 7.13 – PANORAMA(SLED2) (QRC1  $\cup$  QRC3) : Bilan global des expérimentations

Les histogrammes suivants présentent un relevé des marges de temps (Figure 7.14 a)) et des marges d'énergie (Figure 7.14 b)) pour les 8 expérimentations réussies. On peut noter une variabilité somme toute assez faible ce qui semble indiquer une certaine maîtrise globale du déroulement de la mission. La valeur moyenne de marge de temps s'établit à 69.6 s et celle de la marge d'énergie à 0.25 Wh. On peut s'étonner de la variation importante observée pour l'expérimentation 3. A l'analyse, l'augmentation de la marge de temps est induite par une erreur importante de localisation de la méthode LZA qui réduit la taille d'une activité lente. De fait l'erreur d'estimation de pose est sans aucun doute importante. Mais la méthode LZA s'est ensuite correctement recalée ce qui a permis d'achever la mission.

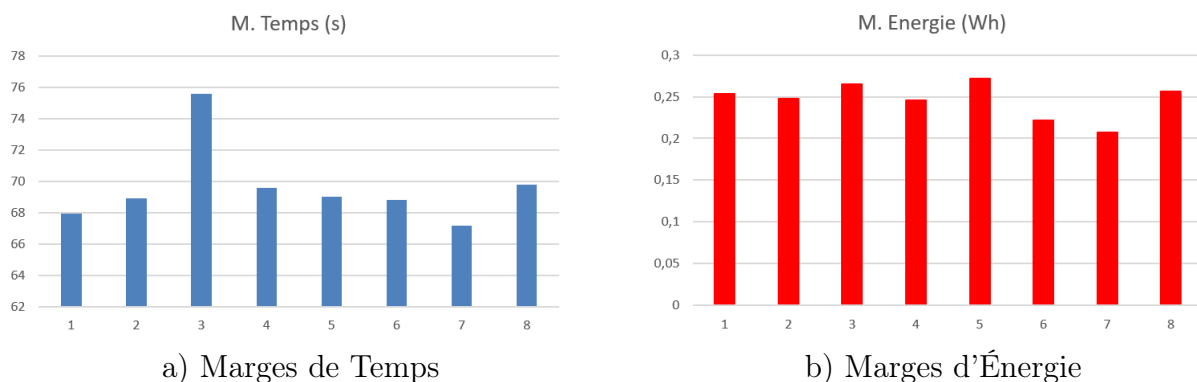


FIGURE 7.14 – Marges expérimentales pour les missions réussies

Globalement à l'analyse de ces différentes expérimentations on peut conclure que l'approche PANORAMA(SLED2) peut permettre de conduire des missions à garantie de performance dans la limite de validité des modèles utilisés bien évidemment.

### 7.4.3 Conclusion

L'instrumentation de l'environnement à l'aide de marqueurs horizontaux placés au plafond a apporté une garantie de détection dans les zones critiques pour lesquelles nos modèles prédictifs d'incertitude avaient été mis en défaut. La série d'expérimentations faite a permis de réaliser avec succès la majorité des missions de Patrouille en procurant les garanties de Sécurité, Durée, Énergie et d'incertitudes de pose demandées. Si l'on exclut les problèmes matériels et logiciels inhérents au processus expérimental les échecs rencontrés relèvent, d'après nous, de plusieurs phénomènes. D'une part, principalement, ces dysfonctionnements sont dus à la non prise en compte des incertitudes de localisation lors des commutations interactivités. D'autre part, ils peuvent aussi découler, surtout dans notre environnement à fort aliasing perceptuel, d'une validation expérimentale nécessairement locale et donc biaisée de certains des modèles prédictifs d'incertitude des méthodes de localisation utilisées.

## 7.5 Conclusion

Plus globalement, l'évaluation expérimentale de la méthodologie PANORAMA(SLED2) confirme qu'il est possible d'apporter un bon niveau de garantie de performance (Sécurité, Durée, Énergie, incertitude de Pose) pour des missions en environnement connu présentant un fort aliasing perceptuel. L'intégration quantitative de l'incertitude de pose a nécessité un travail important pour établir les modèles prédictifs nécessaires. La recherche de garantie a montré sa sensibilité aux conditions expérimentales (garantie de détection) et impacte significativement les dimensions des pavés d'incertitudes pouvant être recherchés (l'incertitude n'est pas la localisation). Enfin, la prise en compte explicite des capacités réelles de localisation des méthodes utilisées a permis de s'affranchir de la logique énergétique à marge minimale proposée dans PANORAMA(SED) pour s'engager dans une logique de maximisation de la marge énergétique en faisant appel à l'algorithme de Viterbi dans PANORAMA(SLED).

# Chapitre 8

## CONCLUSION GÉNÉRALE

Notre étude s'est focalisée sur les problèmes soulevés par la prise en compte quantitative d'une garantie de pose/localisation en intérieur. Après avoir rapidement rappelé le contexte de nos travaux, nous allons synthétiser dans un premier temps les principaux résultats qui, de notre point de vue, constituent les points saillants et les apports originaux de ce travail. Dans un second temps nous en évoquerons les limitations et nous nous pencherons vers les travaux qui pourraient venir compléter les résultats obtenus. Pour terminer nous chercherons à tirer des enseignements tant au sujet des apports de cette étude que sur les difficultés que nous avons rencontrées pour nous projeter plus avant et évoquer l'intérêt que pourrait avoir la méthodologie PANORAMA pour des missions d'exploration.

### 8.1 Contexte des travaux

Le travail que nous avons présenté s'inscrit dans le cadre du développement de missions robotisées autonomes à garantie de performance. Il fait suite aux travaux développés par [112] au sein de l'équipe EXPLORE qui ont permis à la proposition de la méthodologie PANORAMA. Cette dernière permet, au sein d'un environnement connu, d'allouer et de (re)configurer dynamiquement les ressources matérielles et logicielles d'un robot mobile disposant d'une énergie embarquée restreinte, pour qu'il exécute de façon autonome et en sécurité, une mission de durée limitée.

Le contexte expérimental de nos travaux s'inscrit dans le cadre de la localisation indoor qui s'avère difficile à plus d'un titre. D'une part, contrairement aux missions robotisées réalisées en extérieur, il n'est pas possible de faire appel à un système externe (GPS) capable de localiser la plupart du temps, au sein d'un référentiel absolu, la position d'un véhicule, avec une précision connue. D'autre part, les constructions humaines sont bien connues pour constituer des environnements fortement structurés, composés d'une succession de couloirs rectilignes, pauvres en amers discriminants et donc propices à l'émergence d'aliasing perceptuel perturbant fortement la capacité de localisation des systèmes robotisés.

La localisation est une fonction indispensable à tout système robotisé. Les méthodes de localisation constituent très certainement l'un des champs disciplinaires les plus prolifiques de la robotique. S'appuyant sur des bases théoriques multiples (équations de la cinématique et de la dynamique, théorie probabiliste, ensembliste, etc.), répondant à des problèmes variés (localisation locale, globale, kidnapping), faisant appel à des capteurs de différentes natures (proprioceptif, extéroceptif) aux caractéristiques fluctuantes, au sein d'environnements connus ou non, il existe actuellement un très large corpus de méthodes de localisation. L'objectif de nos travaux n'était pas tant de caractériser l'incertitude de localisation de méthodes adressant des



problématiques locales et globales, que de proposer une démarche méthodologique permettant de réaliser cette caractérisation.

Synthétisons à présent les principaux résultats présentés dans ce manuscrit.

## 8.2 Principaux résultats

De nombreux travaux cherchent à évaluer les performances de nouvelles méthodes de localisation. Bien évidemment la précision (ou l'incertitude) affichée est souvent évaluée et parfois comparée à celle d'autres approches. Cependant, si l'on exclut les méthodes de positionnement indoor (*Indoor Positioning System*) en pleine expansion actuellement, et qui pour la plupart ne permettent pas de garantir une précision compatible avec les exigences de la robotique (hors UWB (*Ultra WideBand*)), il n'existe pas, à notre connaissance d'état de l'art analysant le niveau de précision (incertitude) affiché par les principales méthodes de localisation développées en robotique. On peut donc sans doute considérer que les quelques pages que nous y consacrons dans ce manuscrit et en annexe présentent un intérêt certain. Nous y analysons rapidement, au travers d'un ensemble de publications assez récentes, couvrant les principales classes de méthodes de localisation, le niveau de précision affiché, au regard d'un ensemble de critères : Caractéristiques de l'environnement d'expérimentation, moyens d'obtention de la réalité terrain, outillage de l'environnement, etc... Notre analyse montre globalement que les méthodes sont étudiées à l'aide de trajectoires simples dans des conditions expérimentales supposées optimales (ce qui n'est absolument pas le cas dans la réalité) sur un espace souvent limité ou contraint (suivi de route/couloir). L'obtention de la réalité terrain est une problématique difficile qui est finalement rarement abordée précisément alors qu'elle est nécessaire à l'évaluation des performances de localisation d'une méthode et qu'elle impacte de fait les incertitudes affichées. L'utilisation presque exclusive d'un contrôle en boucle ouverte masque les problèmes de contrôle que pourraient présenter les méthodes étudiées, en boucle fermée, en présence d'aliasing perceptuel. L'incertitude de cap est très rarement étudiée alors que son influence est importante sur la qualité du suivi de trajectoire lorsque sa fréquence de rafraichissement est faible. Globalement les erreurs moyennes affichées peuvent atteindre quelques dizaines de centimètres mais sans s'intéresser à la garantie de Localisation et en ne commentant que très rarement la présence d'extrema.

Ce document a aussi permis de proposer, d'évaluer et d'enrichir une nouvelle méthode de localisation globale LZA, qui peut être utilisée au sein d'un environnement connu. C'est une méthode non probabiliste, basée sur les apparences, qui s'appuie sur l'identification, au sein d'une grille, de signatures caractéristiques de la pose du robot, établies à partir de relevés extéroceptifs à 360° équirépartis. L'analyse en simulation a mis en évidence que, comme d'autres approches, la localisation s'avèrerait difficile dans certaines zones de l'environnement essentiellement en raison de l'aliasing perceptuel. Par ailleurs l'expérimentation dans sa forme première a montré une instabilité dans le temps des valeurs renvoyées. De façon à palier ce problème, sans pour autant renoncer à la nature globale de LZA, un algorithme filtrant le phénomène d'instabilité et conditionnant les dimensions de la fenêtre de recherche à l'évolution de la dérive odométrique a été proposé. Les résultats expérimentaux confirment pratiquement le modèle d'incertitude intuitif proposé.

La prise en compte explicite de l'axe localisation au sein de la méthodologie PANORAMA a aussi mis en exergue une problématique qui n'avait pas été jusqu'alors abordée pour la prise en compte des autres axes de performance : la corrélation pouvant exister entre 2 activités successives. Jusqu'alors le choix d'une alternative d'implémentation, au sein d'une activité, était

indépendant de celui effectué au sein de l'activité précédente. Ce n'est plus le cas pour l'axe localisation car, pour pouvoir être opérationnelle au sein d'une activité, certaines méthodes de localisation doivent préalablement être initialisées de façon concomitante à une autre sans pour autant y être effectivement utilisée.

Notre recherche de garantie de localisation nous a conduit assez naturellement à choisir une démarche intervalliste pour établir nos modèles d'incertitude. La plupart des travaux faisant appel à de la localisation ensembliste n'évalue qu'un pavé d'incertitude, associé aux mesures capteurs qui viennent d'être effectuées. Peu d'études cherchent à proposer des approches ensemblistes prédictives [195], et aucune, à notre connaissance, ne s'inscrit dans une logique couplant la sélection des ressources matérielles et logicielles à une garantie de localisation recherchée. Notre proposition s'appuie quant à elle largement sur une démarche expérimentale pour établir les modèles d'incertitude prédictifs qui sont utilisés au sein de l'approche PANORAMA pour sélectionner, tout au long d'une mission, les méthodes de localisation à même de garantir la performance de localisation imposée. Ainsi, tout comme pour les autres axes de performance déjà étudiés (Energie, Durée) nous disposons de critères objectifs pour pouvoir sélectionner la méthode de localisation la mieux à même de satisfaire les contraintes imposées par l'utilisateur. Cette objectivation nous a permis d'abandonner le postulat retenu dans [112] (*une méthode de localisation est d'autant plus performante qu'elle consomme de l'énergie*) pour nous engager dans une logique de maximisation de la marge énergétique.

Pour répondre à cet objectif de maximisation de marge énergétique nous avons fait appel à l'algorithme de Viterbi. Ce dernier, bien connu en programmation dynamique, est parfaitement adapté au problème posé, tant dans la représentation des données, que pour son efficacité d'exécution. Employé en fin du processus de garantie de performance de PANORAMA, il permet de prendre en compte l'axe Énergie et de maximiser la marge correspondante de façon à être moins sensible aux variabilités de la mission (évitements d'obstacle, variation de trajectoire) et aux approximations des modèles énergétiques utilisés que dans l'approche proposée dans [112]. Sa rapidité de résolution permet aussi de pouvoir envisager l'utiliser "on-line" lors d'une mission, en présence de défaillances matérielles et/ou logicielle, pour déterminer un SMD alternatif.

L'ensemble de ces travaux a permis de tester avec succès l'approche de garantie de performance proposée sur une mission de Patrouille utilisée comme exemple fil-rouge. Les garanties SED et SLED ont tout d'abord été détaillées et comparées en simulation. Les scénarios de mission détaillés obtenus ont mis en évidence que la connaissance experte des zones d'application des méthodes de localisation employées dans la méthodologie PANORAMA(SED) [112] n'était pas à même de garantir ni la qualité de localisation, ni la réelle opérationnalité des méthodes de localisation à déployer.

Une quinzaine d'expérimentations ont ensuite été effectuées pour lesquelles une attention particulière a été portée à l'évolution des marges énergétique, de durée et bien sûr à celle de l'incertitude de localisation. Évidemment nous avons pu observer l'impact de l'instrumentation de l'environnement sur la performance de nos méthodes de localisation. L'absence de garantie de détection des QR-Codes verticaux ( $QRC1 \cup QRC2$ ) a conduit expérimentalement à des échecs de mission non identifiés lors des simulations. Il nous a donc fallu rajouter des amers artificiels au plafond ( $QRC1 \cup QRC3$ ) dans les zones où la garantie de localisation n'était pas atteinte.

Dès lors nos objectifs de performance, tant en Durée, qu'en Énergie et en Localisation ont été satisfaits plus de 50% de nos expérimentations. On dénombre cependant trois principales causes d'échec.

D'une part l'utilisation d'une méthode de localisation inadaptée dans la zone où se trouve réellement le robot en raison d'une commutation d'activité inappropriée et cela malgré le continuum fonctionnel local que l'on peut supposer pour les méthodes de localisation. Ce phénomène est induit par le fait qu'actuellement notre approche n'intègre pas l'incertitude de localisation dans la détermination des limites d'activité.

D'autre part, il arrive aussi malheureusement que la méthode de localisation utilisée soit malgré tout perturbée par le fort aliasing perceptuel de notre environnement ce qui induit une incertitude expérimentale supérieure à celle escomptée. Il ne faut en effet pas oublier que nos modèles prédictifs s'appuient d'une part sur des relevés expérimentaux ne reflétant pas nécessairement toutes les conditions pouvant être rencontrées lors d'une mission. De plus, nos modèles, comme c'est le cas pour les approches intervallistes, excluent une partie des relevés expérimentaux semblant aberrants, mais pouvant être évidemment rencontrés.

Nous avons été évidemment confronté aux problèmes habituels d'origine matérielle rencontrés lors d'expérimentations, comme la déconnexion inopportune de capteurs, ou d'origine logicielle, comme des problèmes d'interface de communication entre le driver d'un capteur et ce dernier.

Enfin, la garantie de Localisation nous a autorisé à aborder la dimension énergétique en faisant appel à une logique de minimisation de la consommation grâce à l'algorithme de Viterbi. Cela a permis, contrairement aux travaux proposés dans [112], de réaliser une mission sans remise en cause du SMD initial tout en confirmant la bonne qualité de notre modèle énergétique.

La confrontation des incertitudes de pose mesurées avec une réalité terrain semble montrer qu'il serait possible, la plupart du temps, pendant le déroulement d'une mission, de parler de garantie de localisation et non plus seulement de garantie d'incertitude. En effet la réalité terrain, lorsqu'elle est accessible, recouvre largement l'incertitude d'estimation mesurée. Ainsi, si l'on a confiance dans les méthodes de localisation utilisées à un moment donné, la garantie d'incertitude peut alors décliner en garantie de localisation.

Le travail présenté souffre cependant d'un ensemble de limitations, plus ou moins pénalisantes, qu'il est parfois possible d'atténuer ou avec lesquelles nous devons composer. Nous allons maintenant les balayer.

## 8.3 Limitations

Évidemment le contexte même de notre travail est limitatif puisque la méthodologie PANORAMA s'inscrit nécessairement au sein d'environnement connu où la trajectoire nominale suivie par le robot est supposée choisie. La maîtrise du chemin à suivre, est une condition nécessaire au déploiement des modèles prédictifs. La connaissance de l'environnement est indispensable pour pouvoir connaître le positionnement des amers géoréférencés, les zones où les capteurs sont inopérants (axes Sécurité, Localisation, Durée), et celles où les méthodes de localisation peuvent être déployées.

Cette acquisition d'informations et de connaissances n'est possible qu'après une analyse longue et fortement empirique permettant de déterminer les performances des capteurs et des méthodes de localisation en fonction de leur contexte d'emploi et parfois même spécifiquement pour l'environnement choisi comme c'est le cas pour la méthode LZA. Le coût à payer pour cette caractérisation est donc assez important en terme d'expérimentations, de traitement et d'analyse de données.

Revenons à présent sur la notion même de garantie de localisation qui soutend l'ensemble de ce travail. C'est un concept fort que nous avons abordé par le biais de la construction de modèles prédictif d'incertitudes de localisation. De ce fait, l'atteinte de la garantie recherchée est subordonnée à plusieurs conditions nécessaires. D'une part, les modèles que nous avons établis reposent sur des postulats qui doivent être vérifiés lors des expérimentations. Cela concerne à la fois les conditions environnementales (luminosité, nature des obstacles, etc.), mais aussi les hypothèses retenues telle que la garantie de détection des QR-Codes. D'autre part, la méthode de localisation sélectionnée à un instant donné doit être à même de fournir une estimation fiable de la pose. Par exemple, comme nous l'avons démontré, la méthode LZA ne peut être utilisée sur l'ensemble de l'environnement. La violation d'une de ces deux contraintes conduit inévitablement à la perte de la garantie de localisation. Donc l'exécution de missions autonomes nécessite d'avoir confiance dans les méthodes de localisation utilisées si l'on veut que l'incertitude puisse se décliner en garantie de localisation. Expérimentalement, ces hypothèses peuvent être violées plus ou moins longuement ce qui peut induire une violation passagère des contraintes de localisation rattrapée par le continuum opérationnel des méthodes, ou à une perte totale de la localisation mettant en défaut l'exécution même de la mission. C'est ce que nous avons observé expérimentalement.

Attardons nous maintenant sur les garanties de localisation que nous sommes à même d'apporter grâce au travail présenté dans ce document. Si l'on a un regard critique on ne peut que constater que celles-ci ne correspondent souvent qu'à des pavés d'incertitudes de plusieurs dizaines de centimètres et cela peut paraître bien important par rapport aux résultats affichés dans de nombreux travaux du domaine. Nous avons déjà souligné ici qu'il était inévitable que le robot ne dispose que d'une connaissance approximative de sa localisation en raison des multiples sources d'erreurs venant perturber la détermination de la pose. Parmi ces sources d'erreurs notre méconnaissance du cap exact joue un rôle majeur sur l'incertitude de position. Par ailleurs il faut aussi rappeler que les incertitudes que nous avons établies ne l'ont été qu'avec des moyens de mesure assez limités de la réalité terrain. Cela impacte nécessairement la précision de nos relevés et par la même surestime très certainement les dimensions des pavés d'incertitudes. De meilleurs résultats auraient sans doute été obtenus s'il avait été possible d'utiliser des capteurs plus performants pour obtenir la réalité terrain comme un système *ViCon* par exemple ou même, de façon moins onéreuse, un mécanisme de positionnement externalisé de type UWB, permettant d'assurer en permanence une évaluation de la localisation du robot, avec une incertitude bornée acceptable. En revanche il est essentiel de faire valoir que, contrairement aux approches probabilistes, nous apportons ici une garantie sur la pose du robot et donc du déterminisme. Il est vrai qu'à y regarder de plus près il nous faudrait nuancer quelque peu ce terme car nous n'avons pris en compte que 90% (environ  $2\sigma$ ) des mesures expérimentales pour éliminer les plus extrémales (*outliers*) et préserver des intervalles de dimensions acceptables. Ce choix est généralement retenu dans les travaux faisant appel aux modèles intervallistes. Ces derniers, en prenant en compte les données aberrantes, conduisent, de par leur nature même, à une vision pessimiste et conservatrice des incertitudes de pose.

Relevons aussi une difficulté expérimentale que l'on rencontre inévitablement en environnement intérieur. Même si le robot doit théoriquement suivre un chemin au centre des couloirs, les erreurs de cap, voire de localisation risquent parfois de le faire dévier en direction des murs. Dès lors les distances de sécurité mises en place pour qu'il ne les percute pas vont le faire entrer en évitement d'obstacle et cela d'autant plus vite que le couloir est étroit. Sa trajectoire va alors s'éloigner d'une ligne droite jusqu'à ce qu'il rejoigne le chemin planifié. La qualité de nos modèles odométrique et énergétiques établis pour des trajectoires rectilignes va en être impactée, même si ces écarts sont en général absorbés par les marges de performance associées.

Pour terminer le tour d’horizon des limitations qu’il est possible de pointer du doigt nous allons maintenant nous tourner sur la méthodologie PANORAMA. Déjà dans les travaux de [112] nous avons souligné que la projection des contraintes associées aux performances induisait une multiplication des activités. Ce phénomène a été très fortement accentué par la prise en compte de l’axe localisation où chaque méthode peut induire ses propres contraintes en fonction des capteurs mobilisés et de ses propres performances au sein de l’environnement (accroissement de l’ordre de 20% pour la mission de patrouille étudiée). Si cet émiettement en tant que tel n’est pas inquiétant car les algorithmes employés (Bennour / Viterbi) sont à même de traiter sans problème ce niveau de complexité, cela met en évidence deux écueils. Le premier, et non des moindres, est la difficulté d’automatisation du processus d’élaboration du SMD car ce dernier doit intégrer une grande quantité d’informations de nature très différente. C’est un défi à relever surtout lorsque l’on cherchera à s’inscrire dans une logique d’exploration. Le second est que la projection des multiples contraintes à prendre en compte peut faire émerger des activités de dimensions (durée) très réduites. Il faudra donc s’interroger sur la gestion de ces fines sections de mission face aux incertitudes que nous avons sur leurs limites. Nous reviendrons sur ces deux points un peu plus loin.

Pour terminer ce manuscrit, nous allons maintenant présenter les perspectives qui nous semblent envisageables pour améliorer ou étendre les résultats obtenus, et pour adapter la démarche PANORAMA à d’autres contextes.

## 8.4 Perspectives

La méthode de localisation globale LZA que nous avons détaillée affiche des résultats intéressants même au sein de l’environnement étudié présentant un fort niveau d’aliasing perceptuel. Cependant, faute de temps, nous n’avons pu la comparer, dans ce même contexte, avec d’autres méthodes de localisation. Nous envisageons donc très prochainement de mettre en œuvre une méthode globale de localisation par filtre particulière, tant en simulation qu’en expérimentation, pour en analyser les performances et les comparer avec celles d’LZA. Par ailleurs il nous semble que cette dernière, de part son déterminisme et son caractère global, pourrait s’intégrer avantageusement dans la mise en œuvre d’une méthode de localisation par filtre de Kalman en fournissant un positionnement global garanti qui viendrait guider la sélection probabiliste de la pose du robot.

Même si l’expérimentation est irremplaçable pour déterminer les performances d’un système robotique, la simulation peut se révéler un allier de poids pour faciliter la caractérisation, au sein d’un environnement connu, des méthodes de localisation et en alléger la dimension expérimentale très chronophage. Dans nos travaux, nous y avons déjà fait appel pour l’étude de LZA où des simulations massives, en statique (robot arrêté) ont été mises en œuvre. Nous en avons tiré des informations pertinentes sur les zones d’éligibilité de cette méthode. De même dynamiquement nous avons aussi employé une simulation massive pour déterminer les paramètres permettant de stabiliser le comportement d’LZA. Il nous semble donc intéressant de poursuivre dans cette direction en lançant des campagnes massives de simulations pour chercher à paramétrer et caractériser une méthode de localisation.

Ce type de démarche faisant appel à une masse importante de données est déjà largement employé dans le cadre du Machine Learning. Dans notre cas on peut imaginer faire évoluer virtuellement, un robot selon des trajectoires aléatoires mais connues, en faisant appel à des modèles capteurs et environnementaux réalistes. Il serait alors possible d’établir pour l’environnement choisi, après une campagne massive de simulations, une cartographie d’emploi de la

méthode de localisation évaluée ainsi qu'une première estimation de son modèle d'incertitude et des paramètres l'optimisant.

La méthode PANORAMA, et particulièrement sa dimension garantie de Localisation, peut apporter une aide pertinente pour déployer des robots en environnement intérieur. Elle permet en effet de sélectionner les méthodes les plus à même d'apporter les garanties de localisation recherchées. Mais elle peut aussi participer à l'instrumentation de l'environnement en guidant le placement des amers artificiels en fonction des méthodes retenues et des performances attendues comme cela a été le cas pour le positionnement des QR-Codes de QR3 au plafond.

L'intégration de l'axe de performance Localisation au sein de l'approche PANORAMA souffre actuellement d'une fragilité incontestable qu'il faudra prendre en compte à l'avenir. Le découpage d'une mission repose sur une décomposition métrique supposée exacte de cette dernière en activités. Dès lors comment savoir que l'on passe effectivement d'une activité à une autre lorsque l'on prend en compte les incertitudes de localisation ? Il faudra sans aucun doute s'engager vers une prise en compte conservatrice de ces contraintes pour toutes les méthodes éligibles d'une activité. Ce questionnement rejait sur le problème lié à la prise en compte des activités courtes qui verront leurs dimensions se réduire encore. Quoiqu'il en soit on ne peut aussi que constater à la vue des résultats expérimentaux obtenus que, même si ces incertitudes sur les limites d'activité ne sont pas actuellement considérées dans la méthodologie PANORAMA, les missions sont généralement exécutées en satisfaisant les contraintes de performance imposées. En effet il est évident que, contrairement à ce qui parfois transparait dans notre approche, une méthode de localisation ne cesse pas d'être opérationnelle instantanément lorsque le robot franchit une certaine position au sein de l'environnement. Il existe fort heureusement un continuum fonctionnel qui permet de préserver, dans une certaine mesure, la validité du découpage obtenu.

Par ailleurs le traitement de l'axe Localisation qui a mis en évidence la présence de zone de transition inter-activités nous a amené vers une réflexion plus large sur la notion d'activité. En effet dans les travaux qui avaient posé les bases de la méthodologie PANORAMA, lors de son exécution, une activité n'était composée que des schémas de contrôle sélectionnés et fonctionnellement nécessaires, paramétrés avec des coefficients constants (vitesse/fréquence par exemple). Dorénavant, la prise en compte de l'axe Localisation démontre qu'il faut aussi ajouter au sein des schémas opérationnels d'une activité des schémas qui ne sont pas fonctionnellement utilisés. Par ailleurs, si l'on pousse un peu plus loin la réflexion, l'hypothèse de commutation instantanée des paramètres de configuration des schémas de contrôle lors du passage d'une activité à une autre ne peut être maintenue que dans une certaine limite. Par exemple, actuellement, les gradients de vitesse étant faibles, on suppose justement, que le contrôleur bas niveau prend en charge efficacement ces variations de consigne, et que la durée de cet état transitoire est courte. Cette hypothèse ne serait plus valable avec un robot évoluant à des vitesses supérieures où les gradients seraient bien plus importants. Dans ce cas, entre 2 activités à paramètres constants, il faudrait introduire une activité supplémentaire assurant une transition continue et contrôlée des paramètres modifiés. Ainsi donc nos travaux sur l'axe Localisation nous ont amené à revoir et à adapter notre vision de la notion d'activité.

En l'état, l'affectation des ressources matérielles et logicielles repose sur l'utilisation des algorithmes de Bennour et de Viterbi, décomposant ainsi le problème multicritères (multi-axes de performance) de garantie de performance en plusieurs sous-problèmes (mono axe de performance) pris en compte successivement pour les dimensions Durée et Énergie. Cette démarche a démontré sa capacité à traiter des problèmes de très forte complexité dans un contexte temps réel et embarqué. Il serait tout de même intéressant d'évaluer une démarche plus globale en

faisant appel exclusivement à l'algorithme de Viterbi. En effet, ces 2 dimensions de la performance cherchent à maximiser la marge qui leur est associée. Il est donc envisageable de définir une fonction de coût unique qui prendrait en compte à la fois l'aspect Durée et Énergie.

Enfin, dans ce manuscrit, nous avons cherché à compenser globalement les limitations de chacune des méthodes de localisation étudiée en lui en adjoignant d'autres pour obtenir ainsi des approches plus performantes. Ainsi KIN-ARUCO est devenue KINODO puis COMPODOKIN. Il serait possible de décomposer l'axe de performance Localisation en 3 indicateurs indépendants associés aux incertitudes selon  $X_s$ ,  $Y_s$  et  $\theta$ . La méthodologie PANORAMA composerait alors automatiquement toutes les méthodes disponibles de façon à satisfaire indépendamment chacun des objectifs de Localisation imposés. Cela impacterait sans doute la consommation énergétique et induirait un surcoût en temps de calcul mais permettrait une composition optimale des méthodes de Localisation.

Pour clore notre réflexion nous allons prendre un peu de hauteur et nous interroger sur les cadres applicatifs qu'il est possible d'envisager pour la méthodologie PANORAMA et la mise en œuvre de missions autonomes à garantie de performance.

Tout d'abord les travaux présentés dans [112] et dans ce document ont, nous semble-t-il, démontré l'intérêt de l'approche proposée dans le cadre de missions au long cours, en environnement connu, nécessitant une gestion précise de la dimension énergétique. Ils ont aussi pointé son utilité pour renforcer la tolérance aux fautes [114] des missions exécutées. A notre connaissance, peu de travaux s'intéressent au niveau Mission et sont à même de se projeter vers une garantie multicritères. PANORAMA pourrait donc parfaitement trouver des applications dans le domaine de la surveillance de bâtiments, de bases totalement automatisées comme pour celles de certains sites d'extraction pétrolière [222] ou d'autres dédiées à l'analyse de la biodiversité marine. Cela nécessiterait cependant un important effort de développement en dotant PANORAMA d'une réelle capacité de génération en ligne des Scénarios de Mission Détaillés, et en l'intégrant plus largement au sein d'une architecture logicielle tolérante aux fautes [101].

Mais l'intérêt de PANORAMA ne saurait se limiter exclusivement à la surveillance et cette approche pourrait se révéler particulièrement utile lors de missions d'exploration. En effet ces dernières demandent, la plupart du temps, à ce que le système robotisé revienne à son point de départ, ou, pour le moins, à un endroit où il sera possible de le récupérer. Dès lors se pose la question du moment où l'on doit décider d'achever la phase d'exploration pour engager celle de retour. Pour des raisons énergétiques évidentes cette dernière doit nécessairement s'effectuer au sein d'un environnement connu. Dans ce contexte, l'approche PANORAMA pourrait se révéler un allié des plus utile en permettant la sélection du chemin et des ressources matérielles et logicielles les mieux à même de maximiser la surface explorée tout en apportant des assurances sur la capacité du système à être récupéré. Il est cependant impossible de pouvoir disposer de modèles d'incertitude de localisation évalués explicitement dans l'environnement exploré. Par ailleurs il est déjà difficile de mettre en œuvre des procédures d'évaluation de ces modèles en environnement terrestre d'intérieur, que dire des obstacles qu'il faudrait surmonter pour les établir au sein d'un environnement difficile (marin par exemple). Dès lors, pour prendre en compte l'axe de Performance Localisation il nous semble qu'il faudrait s'orienter vers une stratégie mixte. D'une part établir des modèles d'incertitude de localisation quantitatifs couplant théorie et expérimentations, pour les méthodes faisant appel à des capteurs proprioceptifs (Odométrie, IMU). D'autre part s'appuyer sur l'expertise Humaine des spécialistes du milieu pour définir le contexte environnemental valide de ces modèles de localisation ainsi que l'incertitude qui leur est attachée.

Il va sans dire que la méthodologie PANORAMA étant une démarche d'allocation de ressources au sens large, elle pourrait aussi être utilisée pour coordonner efficacement une flotte de robots hétérogènes, présentant des rôles recouvrants, évoluant de concert. PANORAMA serait donc employée pour gérer globalement le comportement de la flotte, mais aussi localement au niveau de chacun des robots.

Gageons que la méthodologie PANORAMA et ses futures évolutions contribuera à ce que l'équipe EXPLORE puisse atteindre son objectif de réalisation de missions autonomes, mono ou multi-robots, tolérantes aux fautes, à garantie de performance, en environnement difficile connu ou exploré.





# Annexe A

## Énergie : Tables et abaquages des puissances consommées sur la batterie du robot

Les informations présentées ici sont extraites de la thèse [112] qui a proposé la méthodologie PANORAMA. Les valeurs indiquées ont été obtenues expérimentalement sur le Pioneer P3-DX utilisé.

### A.1 Puissance consommée : Modèle utilisé

Puisque nous nous intéressons ici à la batterie robot, l'Equation A.1 ne nécessite pas la prise en compte des ressources logicielles, mais seulement des ressources matérielles impliquées dans le déplacement du robot par le Schéma de Contrôle (SC) sélectionné.

$$P(SC) = \alpha_1 P_{R_{Motion}}(v) + \alpha_2 \cdot P_{R_{US}}(f) + \beta_1 \cdot P_{R_{Kinect1}} + \beta_2 \cdot P_{R_{Kinect2}} + k_1 \cdot \beta_3 \cdot P_{R_{Laser}} \quad (A.1)$$

Où  $k_1 \in \{0, 1, 2\}$  désigne le nombre de lasers utilisés.  $\alpha_i$  ainsi que  $\beta_i$  sont des coefficients booléens indiquant si la ressource matérielle correspondante est utilisée ou non.

### A.2 Ressources matérielles : Table des puissances consommées

La table suivante présente les puissances instantanées statiques, c'est à dire indépendantes de tout paramètre, consommées sur la batterie robot. On remarquera que la Kinect 2 consomme environ 3 fois plus que la Kinect 1.

Composant	Puissance (W)
Contrôleur robot	$P_{R_{Controller1}} = 2.67$
Kinect 1	$P_{R_{Kinect1}} = 2.82$
Kinect 2	$P_{R_{Kinect2}} = 10.12$
Laser	$P_{R_{Laser}} = 2.34$

TABLE A.1 – Puissance instantanée consommée

La puissance associée au contrôleur du robot désigne la puissance nécessaire à l'alimentation des cartes électroniques et du wattmètre embarqué du robot.

## A.3 Puissance et énergie de déplacement

### A.3.1 Modèle de la puissance de déplacement

Le système d'équations suivant définit le modèle de puissance instantanée de déplacement d'un robot Pioneer P3-DX à une vitesse constante  $v$ .

$$\begin{cases} P_{BR_{Motion}}(v) = 2.67 & \text{si } v = 0 \\ P_{BR_{Motion}}(v) = 6.25 \cdot v^2 + 9.79 \cdot v + 3.66 & \text{si } v \neq 0 \end{cases} \quad (\text{A.2})$$

### A.3.2 Énergie de déplacement

Comme  $v = d/\Delta T$  où  $d$  représente la distance parcourue par le robot à une vitesse constante  $v > 0$  pendant  $\Delta T$ , il est possible en utilisant les équations précédentes de déterminer les abaques de la Figure A.1 représentant l'énergie consommée par le robot pour parcourir une distance donnée à une vitesse constante connue.

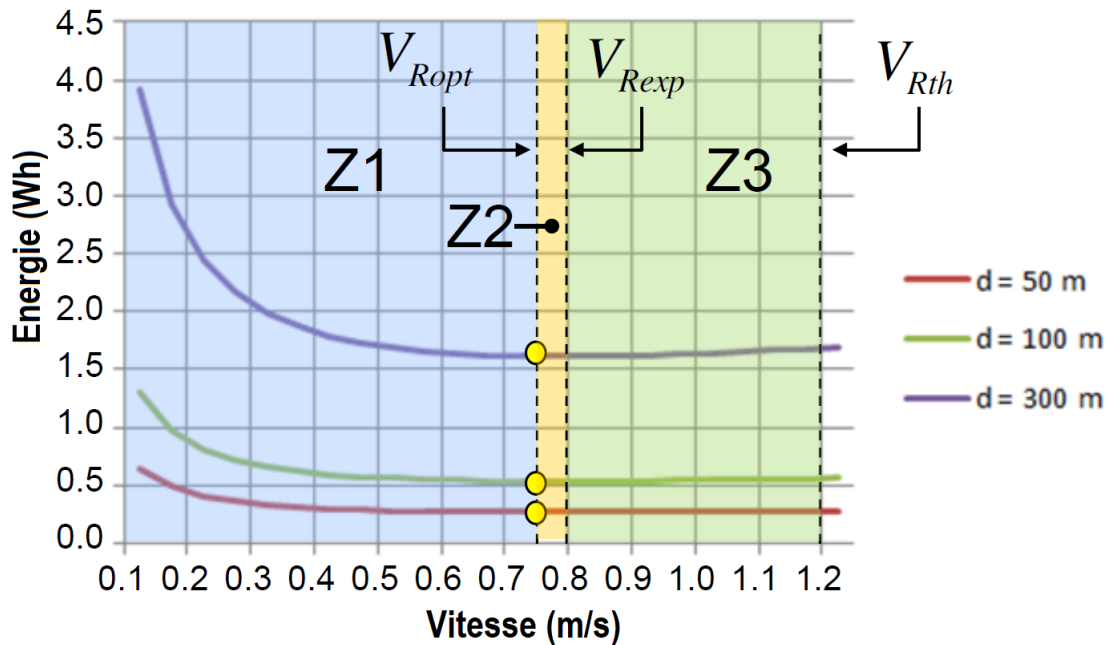


FIGURE A.1 – Énergie de déplacement du robot pour différentes distances

# Annexe B

## Localisation : Tables de synthèse bibliographique

La problématique de localisation est vaste et complexe. Les tables présentées ici ne prétendent pas faire une analyse complète des travaux relevant de la problématique de localisation mais seulement présenter un ensemble d'études représentatives de ce domaine se focalisant sur les performances et conditions expérimentales affichées.

Cette annexe présente l'ensemble des sources utilisées. Elle reprend et complète l'analyse effectuée au cours de l'étude bibliographique présentée dans le chapitre 3 avec les mêmes notations. L'objectif n'est pas de détailler les méthodes présentées mais d'identifier leurs principes ainsi que les performances affichées et leurs conditions d'obtention. Les tables de cette annexe ont toutes la même structure. La première colonne référence la source de l'étude, la deuxième liste les méthodes utilisées en cas de comparaison. Les trois colonnes suivantes caractérisent les expérimentations effectuées (intérieur, extérieur, dimensions, nature des trajectoires). Les performances affichées sont alors synthétisées au sein de la colonne Perf. Pour terminer les deux dernières colonnes abordent les moyens d'obtention de la Réalité Terrain (RT) et le contrôle employé s'ils sont évoqués.

### B.1 Méthodes de localisation locale

Les Tables suivantes présentent des méthodes de localisation locale. Cette classe de méthodes de localisation semble être la plus abordée dans la littérature sans doute car les méthodes globales s'avèrent souvent plus délicates à mettre en œuvre et à caractériser.

La Table B.1 s'intéresse aux travaux expérimentés sur des environnements de dimensions réduites. Ils affichent des incertitudes faibles (de 0.2 mm à 5 cm) mais qu'il faut corrélérer aux dimensions de l'environnement. Les trajectoires suivies sont simples. Les méthodes d'obtention de la réalité terrain sont très variées. Le contrôle utilisé n'est jamais évoqué. On peut donc supposer que sur des environnements aussi réduits le robot se déplace en boucle ouverte.

A contrario la Table B.2 s'intéresse aux méthodes appliquées dans de vastes environnements donc logiquement en extérieur. Les trajectoires suivies sont nécessairement complexes. Le GPS est très souvent employé pour obtenir une réalité terrain. Ses informations sont généralement fusionnées avec d'autres sources de données. Les distances d'erreur moyennes varient de 9 cm à 9.6 m. En plus de la méthode elle-même, cette grande variabilité dépend également des dimensions de l'environnement choisi et des conditions expérimentales rencontrées (extérieur). Le contrôle n'est pas évoqué.

La Table B.3 fait référence aux méthodes appliquées en 3 dimensions. Ces environnements extérieurs restent de dimensions modestes. Les trajectoires sont parfois considérées uniquement en 2D. Les erreurs moyennes vont de 5 cm à 1.5 m. Là encore peu ou pas d'informations sont disponibles sur l'obtention de la réalité terrain ou la nature du contrôle.

Pour terminer les Tables B.4 et B.5 regroupent les environnements de dimensions médianes. Ils représentent la majorité des expérimentations. Les environnements sont la plupart du temps en intérieur que l'on parcourt en suivant des trajectoires souvent simples ou en longeant des couloirs. Étonnement, la question de l'aliasing perceptuel, pourtant centrale dans la localisation en milieu structuré, n'est jamais évoquée. Les erreurs moyennes observées vont de 10 cm à environ 1 m. La réalité terrain est rarement évoquée et, lorsqu'elle l'est, est obtenue par des mesures externes. Le contrôle est comme souvent rarement précisé.

En conclusion, les méthodes locales ont donné lieu à de nombreuses publications expérimentées sur une grande variété d'environnements affichant un large variabilité de dimensions. Sauf cas particuliers, les erreurs/incertitudes moyennes affichées restent supérieures à une dizaine de centimètres. L'incertitude de cap est rarement étudiée. Étonnement, les questions de la réalité terrain, des conditions d'expérimentation, de la nature du contrôle employé ou de l'aliasing perceptuel en environnement intérieur sont souvent peu commentés voire totalement passés sous silence.

TABLE B.1 – Méthodes de localisation Locale sur un environnement de dimensions réduites

Source	Méthodes	Test	Environnement	Trajectoire	Perf.	R.T.	Contrôle
YoonGu 2011 [234]	Localisation avec Amer au plafond et RF	Expérimental	Environnement vide 1.6 x 1.6 m Statique	Statique	Erreur moyenne : 2.36 cm	Positionnement manuel	?
	Time-Difference-of-Arrival (TDOA)				Erreur moyenne : 5.92 cm		
Loevsky 2021 [149]	Marqueurs avec caméra omni-directionnelle	Expérimental	Environnement vide 1.6 x 1.6 m	Simple ligne droite	Intervalle d'erreur +/-5 cm latéral	Trajectoire de référence	?
			Trajectoire Rectiligne 1m		Non étudiée		
Mikulova 2017 [165]	KF + Odométrie + Amers	Expérimental	Environnement vide de 1500 x 1000 unités. (supposé en mm)	Simple : Trajectoire en S	Étude qualitative de la fréquence de localisation et forme de la trajectoire obtenue	Camera externe observant un damier	?
			Espace dégagé 0.5 x 0.5 m		Erreur distance parcourue 3.96 cm Erreur orientation finale -2.26°		
Cho 2012 [59]	SCT	Expérimental	Surface plane et vide 2.5 x 2.5 m	Simples : Droites, cercles et S	Distance Erreur [1, 3] cm	Tracé au sol par le robot	?
Choi 2010 [60]	Correction par amer magnétique.	Expérimental	Espace de test instrumenté Trajectoire d'environ 60 cm Le robot commence et finit sur un amer.	Ligne droite A/R	Erreur Moyenne X : 0.23 mm Y : 0.66 mm théta : 0.31°	?	?.

TABLE B.2 – Méthodes de localisation Locale sur un environnement de grandes dimensions

Source	Méthodes	Test	Environnement	Trajectoire	Perf.	R.T.	Contrôle
Irie 2016 [110]	Localisation carte ville Amers naturels augmentés	Jeu de données	Images sur trottoirs (37) Images sur route (575) Estimation à chaque image Position approximative connue	-	Distance Erreur [0, 2] m Moyenne = 0.64 m Orientation Erreur [0, 20°] Moyenne = 6.5°	Position donnée (trottoir) ou GPS (sur route)	-
					Distance Erreur [0, 2] m Moyenne = 1.1 m Orientation Erreur [0, 20] en ° Moyenne = 8.7°		
	Estimation par correspondance de bord de route	Expérimental	Enregistrement des mesures sur une trajectoire en ville de 150 m	Complexe	Distance Erreur moyenne 0.6 +/- 0.13 m	GPS cinématique	?
					Distance Erreur moyenne 1.2 +/- 0.78 m		
Distance Erreur moyenne 9.58 +/- 20.7 m							
Distance Erreur moyenne 1.33 +/- 0.1 m							
Gonzales 2011 [93]	Compas visuel + Odométrie visuelle	Expérimental	Environnement extérieur 15 x 60 m	Simple : Rectangle	Erreur : [0.5, 6] m	GPS différentiel et compas magnétique	?
					Odométrie visuelle seule		
	Compas visuel + Odométrie visuelle	ODO Visu + Compas Visu [3-10] m					
	Odométrie visuelle	ODO visuelle : [0.5, .85] m					
Aynaud 2017 [19]	AKF	Expérimental	Environnement extérieur S de 70m d'amplitude répétés tous les 40 m	Complexe : Trajectoire en S	ODO : [0.5, 40] m	?	?
					Distance [0.02, 0.32] m Moyenne = 0.091 m Médiane : 0.078 m		

TABLE B.3 – Méthodes de localisation Locale en environnement 3D

Source	Méthodes	Test	Environnement	Trajectoire	Perf.	R.T.	Contrôle
Luo 2019 [152]	OMRDL	Simulation	Piscine 8 x 4 x 5 m	Trajectoire 2D Rectangle, droite Cercle	Erreurs moyennes : Rectangle : 20.4 mm Droite : 72.4 mm Cercle : 22.5 mm	?	?
		Expérimental	Piscine 8 x 4 x 2.5 m	Trajectoire 2D Droite, 1/2 cercle	Erreur max rectiligne : 45 mm Courbe : 87 mm	?	?
Jung 2016 [122]	Localisation par amers artificiels + amers structure	Expérimental	Piscine 8 x 8 x 6.5 m	Trajectoire 3D faisant le tour d'une structure	Distance d'erreur Erreur moyenne : 1.43 m	Points de passage	?



TABLE B.4 – Méthodes de localisation Locale 1

Source	Méthodes	Test	Environnement	Trajectoire	Perf.	R.T.	Contrôle
An 2018 [9]	AKF	Expérimental	Extérieur, chemin goudronnés 100 x 65 m (estimé)	Simple : Rectangle	Distance : [0.61, 4.91] m	-	-
Nagata 2016 [214]	Fusion ODO/Gyro.	Expérimental	Espace dégagé Sol naturel (terre) avec bosses. 30 x 30 m	Simple : Carré	Distance Erreur : Fusion : [0, 5] m	LN100 Erreur fabricant +/-3mm	Téléop.
	Odo.				Distance Erreur: Odo : [0, 43] m		
	Gyro.				Distance Erreur : Gyro : [0, 7] m		
	EKF-GPS				Distance Erreur : EKF : [1, 5] m		
Kim 2016 [131]	Odométrie recalée	Expérimental	Deux salles reliées d'un Couloir 60 x 10 m	Complexe : Exploration des salles et du couloir	Faible Distance Erreur < 1 m	?	?
	Odométrie				Distance Erreur > 30 m		
Fernandes 2015 [78]	EKF extraction d'amers naturel : Portes	Expérimental	Couloir ~ 70 m avec une intersection au milieu 8 A/R rectilignes ~ 1.1 km	Simple : Longue distance Rectiligne	$2 \sigma_x = 0.06$ m Max : 0.08 m $2 \sigma_y = \sim 0.06$ m Max = 0.5 m $2 \sigma_\alpha \sim 3.5^\circ$ Max $10^\circ$	?	?
					Expérimental		

TABLE B.5 – Méthodes de localisation Locale 2

Source	Méthodes	Test	Environnement	Trajectoire	Perf.	R.T.	Contrôle
Fu 2013 [85]	KF ODO/WSN Bornes Zigbee (BDR)	Expérimental	Environnement plat 6 x 6 m avec des bornes tout autour	Simple : Trajectoires en S	Erreur X = [0, 0.11] m	?	?
	Erreur Y = [0, 0.14] m						
	Odométrie						
	Amer Zigbee seul						
	KF simple				Erreur X = [0, 0.15] m		
					Erreur Y = [0, 0.19] m		
					Erreur X = [0, 1.3] m		
					Erreur Y = [0, 1.83] m		
Zhao 2011 [242]	Odométrie monoculaire Flot optique	Expérimental	Extérieur chemin goudronné 10 x 20 m	Complexe : Suivi du chemin goudronné	Estimations données pour des parcours différents non précisés	GPS	?
Gai 2016 [86]	IEKF	Expérimental	Environnement intérieur carrelé 10 x 25 m	Simple : Trajectoire en S	Distance Erreur [0, 0.8] m	?	Téléprog.
Farouc 2011 [20]	Fusion odométrie Gyroscope	Expérimental	Non décrit ! (capteurs proprioceptifs) A priori plan 3 x 3 m	Simple : Carré	Erreur finale X : [-120, 50] mm	Position finale mesurée	?
					Erreur finale Y : [-70, 20] mm		
					Erreur finale X : [50, 130] mm		
	EKF				Erreur finale Y : [10, 60] mm		
	UKF				Erreur finale X : [-20, 25] mm		
					Erreur finale Y : [-20, 10] mm		
Zhang 2018 [241]	Détection d'amers naturels	Expérimental	Espace dégagé 6 x 3.5 m	Partie d'un rectangle	Distance [0, 45] mm	?	?
	Odométrie visuelle			Rectangle	Distance [0, 95] mm		

## B.2 Méthodes de localisation globale

La Table B.6 s'intéresse aux méthodes de localisation globale. On peut remarquer qu'elles se déploient toutes en environnement clos ce qui tend à limiter la taille des expérimentations. On parcourt souvent des couloirs en adoptant donc des trajectoires complexes. Les erreurs maximales affichées sont généralement de l'ordre de plusieurs dizaines de centimètres et peuvent même atteindre plus d'un mètre. Cela peut paraître important dans des environnements de dimensions modestes mais cela peut provenir d'une pose de départ inconnue conduisant la méthode à résoudre initialement un problème de kidnapping. L'incertitude de cap semble être plus souvent prise en considération que pour les méthodes locales. On note la présence de quelques études effectuées exclusivement en simulation, même si cette dernière est largement évoquée dans les articles avant d'aborder la partie expérimentale. Le problème de l'aliasing perceptuel n'est pas du tout abordé alors qu'il est très pénalisant pour les approches globales. Contrairement aux méthodes locales, le mécanisme d'obtention de la réalité terrain est souvent décrit. En revanche la nature du contrôle est toujours aussi rarement abordée.

TABLE B.6 – Méthodes de localisation Globale

Source	Méthodes	Test	Environnement	Trajectoire	Perf.	R.T.	Contrôle
Akai 2018 [10]	CNN	Simulation	20 x 30 m	Trajectoire Complexe Boucle	Distance [0.02, 1.2] m Thêta : [-4, 2] en ° Indice Confiance [0, 1]	Simulation	?
		Expérimental	20 x 25 m	Trajectoire Complexe Boucle	Indice Confiance [0,1]	?	?
Perez 2017 [102]	Grille à découpage flu	Expérimental	Pièce vide de 12.5 x 9 m	Complexe : entrée dans la salle puis parcours au centre	Distance d'erreur : [2, 27] cm Erreur d'orientation : [0, 16°]	Réestimation position par triangulation opportunistes	Contrôle manuel
		Expérimental	15.3 x 42.6 m	Complexe : Couloir coudé Intersection au coude Suivi du couloir	Distance d'erreur : [2, 66] cm Erreur d'orientation [0, 16°]	Réestimation position par triangulation opportunistes	Contrôle manuel
Li 2019 [146]	Grille PTPSM	Expérimental	Couloir 40 x 3 m	125 estimations de pose initiale	Erreur : X [0, 1.9] m Y [0, 1.9] m Thêta : [0.01, 0.09] rad Temps [20 37 ] ms	AT960 Capteur	?
	MCL	Expérimental	6 x 6 m	Environnement type bureau openspace Trajectoire en U dans les couloirs	Erreur : X : [0, 0.112] m Y : [0, 0.122] m  Erreur : X : [0, 0.189] m Y : [0, 0.188] m		
Yilmaz 2019 [233]	SA-MCL	Expérimental	Parcours des couloirs d'un bâtiment 140 x 80 m	Complexe : Parcours de couloir avec intersection	$\sigma_x = 0.21$ m $\sigma_y = 0.09$ m	Amers externes	?
	AMCL				Sigma EX = 0.39 m Sigma EY = 0.32 m		
Zhu 2011 [244]	PFOI	Simulation	Intérieur simulé 120 x 60 m	Data set de couloirs formant un carré	Erreur = f(Nb particules) [2, 17] cm	?	?
	PFI			Parcours de couloirs	Erreur = f(Nb particules) [2, 27] cm		

## B.3 Méthodes de localisation par intervalle

La Table B.7 reprend les éléments abordés dans l'état de l'art présenté dans le chapitre 3 en détaillant simplement les performances obtenues par les méthodes analysées.

TABLE B.7 – Méthodes de localisation par intervalle

Source	Méthodes	Test	Environnement	Trajectoire	Perf.	R.T.	Contrôle
Colle 2013 [63]	Triangulation par set inversion.	Expérimental	Pièce de 9.4 x 6.4 m	Statique	Temps de calcul en fonction de l'incertitude permise: Tps : [0.016, 0.089] s Grain : [0.1, 0.01] m	Placement à la main	-
Guyonneau 2014 [99]	IAL	Expérimental	Environnement type labyrinthe 10 x 4.7 m	Statique	Tps de calcul entre 21-45 s 1 <sup>ère</sup> itération carte totale $\partial X = [18.1, 122.1]$ mm $\partial Y = [16.5, 51.5]$ mm $\partial \theta = [0.55, 12.2]^\circ$	?	-
			Environnement intérieur 20 x 20 m	?	Pour tps de calcul = 28 s $\partial X = 114$ mm $\partial Y = 105$ mm $\partial \theta = 2.8^\circ$		?
			Couloir 26 x 4 m	Simple : Rectiligne	Incertitude finale : $\partial X = 21$ cm $\partial Y = 7.5$ cm $\partial \theta = 3.5^\circ$		?



# Annexe C

## Environnement : Référencement des QR-Codes ARUCO

Les tables suivantes présentent les poses/positions des différents ensembles de QR-Codes évoqués au cours de ce manuscrit. Leurs coordonnées ont été obtenues en croisant des mesures locales réalisées au mètre à ruban, ou le cas échéant, au télémètre laser, avec la carte utilisée pour la méthode LZA. L'orientation, lorsqu'elle est donnée, correspond à l'orientation du QR-Code dans le repère monde en considérant le référentiel défini au chapitre 4.

### C.1 Ensemble Initial : QRC1

Les coordonnées présentées dans la Table C.1 sont celles des QR-Codes initialement prévus pour exécuter la mission de Patrouille. Ils correspondent aux marqueurs utilisés dans [112] qui a servi de référence et de point de départ à nos expérimentations. Ils sont pratiquement tous placés à hauteur de caméra sur des murs saillants. Cela explique la raison pour laquelle, à quelques rares exceptions (amers visibles dans les angles de l'environnement), ces QR-Codes ne sont visibles que frontalement dans un sens de parcours (Aller ou Retour) du chemin. On notera que les indices (ID) 478 et 58 correspondent à ceux des marqueurs placés juste au-dessus des vanes 1 et 2 respectivement.

### C.2 QR-Codes verticaux ajoutés : QRC2

Les coordonnées présentées dans la Table C.2 sont celles des QR-Codes ajoutés suite à la non convergence de PANORAMA-SLED2(QRC1). Ils ont été placés sur des supports mobiles à des emplacements repérés par un marquage au sol. Ces QR-Codes ont été ajoutés dans les zones où PANORAMA-SLED2(QRC1) ne pouvait satisfaire les contraintes de Localisation imposées. Cela induit une forte concentration de QR-Codes au niveau de la salle de repos et de la halle robotique. Par ailleurs, certains marqueurs du couloir 2, initialement employés pour obtenir la réalité terrain lors de nos caractérisations expérimentales, ont aussi été intégrés à QRC2. Visibles latéralement, ils peuvent être utilisés par la méthode COMPODOKIN.

### C.3 QR-Codes verticaux ajoutés : QRC3

Les coordonnées présentées dans la Table C.3 sont celles des QR-Codes ajoutés pour être utilisés par la méthode KIN2ODO. Le modèle d'incertitude établi a permis de judicieusement les placer et les espacer pour espérer atteindre les performances attendues dans ces zones. Il est normal de les retrouver aux emplacements des amers de QRC2 puisque QRC3 vient remplacer

ID	Pose			Visibilité	
	X (m)	Y (m)	$\theta$ (rad)	Vue Aller	Vue Retour
1	26.04	52.27	$\pi/2$	Front	
300	26.35	50.46	$\pi/2$	Front	
47	26.78	42.49	$\pi/2$	Front	
4	28.57	41.44	$\pi/2$	Front	
16	28.57	35.19	$\pi/2$	Front	
3	26.70	35.16	$\pi/2$	Front	
18	28.57	32.49	$\pi/2$	Front	
29	28.59	28.97	$\pi/2$	Front	
478 (V1)	28.49	28.21	$\pi$	Front	
5	28.59	22.63	$\pi/2$	Front	
24	26.70	22.63	$\pi/2$	Front	
30	28.58	16.48	$\pi/2$	Front	
100	26.7	11.79	$\pi/2$	Front	
17	28.58	10.19	$\pi/2$	Front	
10	29.00	1.20	$\pi$	Latéral	Front
6	27.45	0	$\pi/2$	Front	Latéral
7	17.88	2.41	0	Front	
301	7.81	2.44	0	Front	
320	3.67	2.42	0	Front	
23	0.15	0	$\pi/2$	Latéral	Front
8	-1.05	-0.12	0	Front	Latéral
56 (V2)	-0.17	5.35	$-\pi/2$	Front	
31	3.46	2.44	$\pi$		Front
32	5.18	2.40	$\pi$		Front
15	22.11	2.40	$\pi$		Front
25	28.58	7.07	$-\pi/2$		Front
11	28.58	13.35	$-\pi/2$		Front
26	28.59	19.61	$-\pi/2$		Front
12	28.59	25.83	$-\pi/2$		Front
27	28.57	32.17	$-\pi/2$		Front
13	28.58	38.38	$-\pi/2$		Front
28	28.88	46.05	$-\pi/2$		Front
21	29.04	52.17	$-\pi/2$		Front
888	29.04	54.12	$-\pi/2$		Front
14	26.57	32.17	$-\pi/2$		Front

TABLE C.1 – Table des poses des QR-Codes ARUCO de l'ensemble QRC1

ID	Pose			Visibilité	
	X (m)	Y (m)	$\theta$ (rad)	Vue Aller	Vue Retour
20	28.10	57.00	$\pi$	Front	
903	26.8	52.57	$\pi/2$	Front	
930	29.04	50.67	$\pi/2$	Front	
908	26.80	48.97	$\pi/2$	Front	
902	28.8	47.01	$\pi/2$	Front	
916	26.79	47.01	$\pi/2$	Front	
941	26.90	46.03	$\pi/2$	Front	
912	26.80	27.42	$\pi/2$	Front	
917	26.80	2.30	$\pi/2$	Front	
402	24.90	0	$\pi/2$	Latéral	Latéral
940	24.90	2.15	0	Front	
406	20.29	0	$\pi/2$	Latéral	Latéral
944	20.29	2.15	0	Front	
947	20.28	2.15	$\pi$		Front
905	17.88	0.15	0	Front	
411	15.27	0	$\pi/2$	Latéral	Latéral
909	15.27	2.15	0	Front	
416	9.76	0	$\pi/2$	Latéral	Latéral
943	9.76	2.15	0	Front	
952	9.75	2.15	$\pi$		Front
919	7.81	0.2	0		Front
419	5.27	0	$\pi/2$	Latéral	Latéral
915	1.15	0.15	0	Front	
907	0	0.15	0	Front	
904	-1.80	1.10	0	Front	
945	1.14	0.15	$\pi$		Front
901	2.63	0.15	$\pi$		Front
906	5.18	0.20	$\pi$		Front
948	7.80	0.12	$\pi$		Front
949	15.26	2.15	$\pi$		Front
910	22.11	0.15	$\pi$		Front
918	26.65	2.10	$\pi$		Front
914	28.80	4.31	$-\pi/2$		Front
913	26.78	46.05	$-\pi/2$		Front
950	26.8	48.96	$-\pi/2$		Front
951	26.4	52.26	$-\pi/2$		Front
942	27.10	55.25	$-\pi/2$		Front
19	27.70	58.50	$-\pi/2$	Latéral	Front
911	24.00	57.40	0		Front

TABLE C.2 – Table des poses des QR-Codes ARUCO de l'ensemble QRC2



et compléter QRC2. La Table C.3 ne présente pas d'indication de visibilité. En effet la caméra Kinect 2 observant le plafond, les QR-Codes lui font toujours face quel que soit le sens de déplacement du robot. La hauteur de positionnement des amers est en revanche importante dans les transformations permettant de déduire l'estimation de pose du robot, c'est pourquoi elle a été renseignée. Tous les QR-Codes de QRC3 ont été placés suivant la même orientation. C'est pourquoi seule leur localisation-3D est indiquée.

	Localisation-3D				Localisation-3D		
ID	X (m)	Y (m)	Z (m)	ID	X (m)	Y (m)	Z (m)
203	13.30	1.52	2.20	204	14.41	1.52	2.20
205	15.97	1.52	2.20	206	17.61	1.52	2.20
207	18.93	1.52	2.20	208	20.47	1.52	2.20
209	21.97	1.52	2.20	210	23.43	1.52	2.20
211	25.03	1.52	2.20	212	8.03	1.51	2.22
213	5.08	1.51	2.27	214	2.01	1.51	2.22
215	-0.25	1.51	2.22	216	-0.17	3.2	2.78
217	27.54	44.97	2.25	218	27.70	47.76	2.17
219	27.62	50.69	2.17	220	27.70	53.91	2.49
221	27.53	56.90	2.65	222	26.65	1.51	2.22
223	27.53	2.20	2.22	224	10.86	1.52	2.20

TABLE C.3 – Table des localisation-3D des QR-Codes ARUCO de l'ensemble QRC3

# Bibliographie

- [1] Merriau, P.x and Dupuis, Y. and Boutteau, R. and Vasseur, P. and Savatier, X. LiDAR point clouds correction acquired from a moving car based on CAN-bus data, 2017.
- [2] Ohmae, T. and Matsuda, T., and Kamiyama, K. and Tachikawa, M. A Microprocessor-Controlled High-Accuracy Wide-Range Speed Regulator for Motor Drives. *IEEE Transactions on Industrial Electronics*, IE-29(3) :207–211, 1982.
- [3] ISO/TS 15066 :2016. Robots et dispositifs robotiques — Robots coopératifs. Technical report, International Organization for Standardization, 2016.
- [4] DS/ISO 18646-1. Robotics - Performance criteria and related test method for service robots - Part 1 : Locomotion for wheeled robots. Technical report, International Organization for Standardization, 2016.
- [5] ISO 18646-2. Robotics - Performance criteria and related test method for service robots - Part 2 : Navigation. Technical report, International Organization for Standardization, 2019.
- [6] JCGM 200 :2012. International vocabulary of metrology – Basic and general concepts and associated terms (VIM). Technical report, Joint Committee for Guides in Metrology (JCGM), 2012.
- [7] ISO/IEC 25010. System and software quality models. Technical report, International Organization for Standardization, 2011.
- [8] ISO 9283 :1998(fr). Manipulating industrial robots — Performance criteria and related test methods. Technical report, International Organization for Standardization, 1998.
- [9] Jongwoo A. and Jangmyung L. Robust positioning and navigation of a mobile robot in an urban environment using a motion estimator. *Robotica*, 37(8) :1320–1331, 2019.
- [10] Naoki Akai, Luis Yoichi Morales, and Hiroshi Murase. Simultaneous pose and reliability estimation using convolutional neural network and Rao–Blackwellized particle filter. *Advanced Robotics*, 32(17) :930–944, 2018.
- [11] R. Alami and al. An Architecture for Autonomy. *International Journal of Robotic Research*, 17 :315–337, 1998.
- [12] Albagul, A. and Wahyudi, A. Dynamic Modelling and Adaptive Traction Control for Mobile Robots. *International Journal of Advanced Robotic Systems*, 1(3) :149–154, 2004.
- [13] Alonzo, K. Some Useful Results for Closed-Form Propagation of Error in Vehicle Odometry. Technical Report CMU-RI-TR-00-20, Carnegie Mellon University, Pittsburgh, PA, December 2000.
- [14] Aplicaciones de la Vision Artificial. ArUco : a minimal library for Augmented Reality applications based on OpenCV. <https://www.uco.es/investiga/grupos/ava/node/26>, 2020.
- [15] Arkin, R. and Cameron, J. and MacKenzie, D. Multiagent Mission Specification and Execution. *Autonomous Robots*, 4 :29–52, 03 1997.

- [16] Arroyo, R. and Alcantarilla, P. F. and Bergasa, L. M. and Romera, E. Towards life-long visual localization using an efficient matching of binary sequences from images. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6328–6335, 2015.
- [17] Ashokaraj, I. and Tsourdos, A. and Silson, P. and White, B. Sensor based robot localisation and navigation : using interval analysis and extended Kalman filter. In *2004 5th Asian Control Conference*, volume 2, pages 1086–1093, 2004.
- [18] Awad, F. and Shamroukh, R. Human Detection by Robotic Urban Search and Rescue Using Image Processing and Neural Networks. *International Journal of Intelligence Science*, 04 :39–53, 01 2014.
- [19] Claude Aynaud, Coralie Bernay-Angeletti, Romuald Aufrère, Laurent Lequievre, Christophe Debain, and Roland Chapuis. Real-Time Multisensor Vehicle Localization : A Geographical Information System Based Approach. *IEEE Robotics and Automation Magazine*, 24(3) :65 – 74, September 2017.
- [20] Azizi F. and Houshangi, N. Mobile Robot Position Determination. 2011.
- [21] Babineca, A. and Jurisic, L. and Hubinskya, P. and Duchon, F. Visual localization of mobile robot using artificial markers. *Procedia Engineering*, 96 :1–9, 2014.
- [22] Backurs, A. and Tzamos, C. Improving Viterbi is Hard : Better Runtimes Imply Faster Clique Algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 311–321, 08 2017.
- [23] Badino, H. A Robust Approach for Ego-Motion Estimation Using a Mobile Stereo Platform. In *International Workshop on Complex Motion - IWCM 2004*, pages 198–208, 01 2004.
- [24] Baldwin, I. and Newman, P. Laser-only road-vehicle localization with dual 2D push-broom LIDARS and 3D priors. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2490–2497, 2012.
- [25] Bayle, B. *Robotique Mobile*. Cours Ecole Nationale Supérieure de Physique de Strasbourg - Université Louis Pasteur, 2009.
- [26] Beer, J. M. and Fisk, A. D. and Rogers, W. A. Toward a Framework for Levels of Robot Autonomy in Human-robot Interaction. *Journal of Human-Robot Interaction*, 3(2) :74–99, July 2014.
- [27] Bekey, G. A. *Autonomous Robot : From Biological Inspiration to Implementation and Control*. Cambridge, MA : The MIT Press, 2005.
- [28] Bellman, R. On a Routing Problem. *Quarterly of Applied Mathematics*, 16 :87–90, 1958.
- [29] M. Bennour. *Contribution à la Modélisation et à l’Affectation des Ressources Humaines dans les Processus*. Thèse de doctorat, Université de Montpellier, 2004.
- [30] Bennour, M. and Crestani, D. and Crespo, O. and Prunet, F. Computer-Aided Decision for Human Task Allocation with Mono and Multi-Performance Evaluation. *International Journal of Production Research*, 43(21) :4559–4588, Nov 2005.
- [31] Berthomieu, B. and Vernadat, F. Time Petri nets analysis with TINA. In *Third International Conference on the Quantitative Evaluation of Systems, QEST 2006*, pages 123–124, 01 2006.
- [32] Bescos, P. L. and al. *Contrôle de gestion et management*. Guide des techniques et de la décision, 1995.
- [33] Bonsignorio, F. and Hallam, J. and Del Pobil, A. P. Good experimental methodology guidelines. Technical report, Special Interest Group on Good Experimental Methodology in Robotics European Robotics Research Network (EURON), 2008.

- [34] Bonsignorio, F. and Hallam, J. and Del Pobil, A. P. Performance evaluation method of intelligent mobile robot platform for household and similar applications. Technical report, International Electrotechnical Commission, 2012.
- [35] Borenstein, J. and Feng, L. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transaction on Robotics and Automation*, 12(6) :869–880, 1996.
- [36] Bostelman, R. *Performance measurement of mobile manipulators*. Thèse de doctorat, Université de Bourgogne, 2018.
- [37] A. Bourguignon. Peut-on définir la performance ? *Revue Française de Comptabilité*, pages 61–66, 1995.
- [38] Bouvet, C. De l'autonomie au pouvoir d'agir dans les thérapies cognitivo-comportementales. *Pratiques en santé mentale*, 62e année(3) :51–54, 2016.
- [39] Bradshaw, J. and al. Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction. In *International Workshop on Computational Autonomy - Autonomy 2003*, pages 17–39, 07 2003.
- [40] Brandner, C. *Encyclopedia of Neuroscience*, chapter Spatial memory, page 3804–3806. Springer-Verlag, 2009.
- [41] Braynov, S. and Hexmoor, H. . *Agent Autonomy. Multiagent Systems, Artificial Societies, and Simulated Organizations*, chapter Quantifying Relative Autonomy in Multiagent Interaction. International Book Serie, vol. 7, Springer, Boston, MA, 2003.
- [42] Yohan Breux. *Du capteur à la sémantique : contribution à la modélisation d'environnement pour la robotique autonome en interaction avec l'humain*. Thèses, Université Montpellier, Nov 2018.
- [43] Brooks, R. A. A robust layered control system for a mobile robot. *Robotics and Automation*, 2 :14–23, 1986.
- [44] Brown, J. ZBar bar code reader. <http://zbar.sourceforge.net/>, 2010.
- [45] Browning, T. R. Applying the design structure matrix to system decomposition and integration problems : a review and new directions. *IEEE Transaction Engineering Management*, 48 :292–306, 2001.
- [46] Burchardt, A. and Laue, T. and Röfer, T. Optimizing Particle Filter Parameters for Self-localization ;. In *RoboCup 2010 : Robot Soccer World Cup XIV*, pages 145–156. Springer Berlin Heidelberg, 2011.
- [47] Burgard, W. and Fox, D. and Hennig, D. and Schmidt, T. Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids. In *Proceedings of the thirteenth national conference on Artificial intelligence*, volume 2, page 896–901, 03 1996.
- [48] Burguera, A. and Cid, Y. and Oliver, G. Sonar Sensor Models and Their Application to Mobile Robot Localization. *Sensors*, 9 :10217–43, 12 2009.
- [49] Calisi, D. and Nardi, D. Performance evaluation of pure-motion tasks for mobile robots with respect to world models. *Autonomous Robots*, 27 :465–481, 2009.
- [50] Campion, G. and Bastin, G. and Dandrea-Novel, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1) :47–62, 1996.
- [51] Carvalho Filho, J. G. N. D. and Carvalho, E. A. N. and Molina, L. and Freire, E. O. The Impact of Parametric Uncertainties on Mobile Robots Velocities and Pose Estimation. *IEEE Access*, 7 :69070–69086, 2019.
- [52] Cavanini, L. and al. A QR-code localization system for mobile robots : Application to smart wheelchairs. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2017.

- [53] Ceballos, N. D. M. and Valencia, J. A. and Ospina, N. L. *Mobile Robots Navigation*, chapter Quantitative Performance Metrics for Mobile Robots Navigation, pages 485–500. IntechOpen, 2010.
- [54] Cen, G. and Matsuhira, N. and Hirokawa, J. and Ogawa, K. and Hagiwara, I. Mobile robot global localization using particle filters. In *2008 International Conference on Control, Automation and Systems*, pages 710–713, 2008.
- [55] Chabert, G. and Jaulin, L. Contractor programming. *Artificial Intelligence*, 173(113) :1079–1100, Jun 2009.
- [56] Chaîne You Tube EXPLORE. SL2 : Mobile Robot Fast Global Localization. <https://www.youtube.com/watch?v=oAZPaPP606I>, 2019.
- [57] Chatila, R. and Laumond, J. Position referencing and consistent world modeling for mobile robots. In *International Conference on Robotics and Automation (ICRA)*, volume 2, pages 138–145, 1985.
- [58] Chatilla, R. and al. Ethique de la recherche en robotique. Technical report, ALLISTENE - Alliance des Sciences et Technologies du Numérique, 2014.
- [59] Cho, S. and Park, J. and Lee, J. A dynamic localization algorithm for a high-speed mobile robot using indoor gps. *Robotica*, 30(4) :681–690, 2012.
- [60] Choi, B. J. and Kim, B. and all. Magnetic landmark-based position correction technique for mobile robots with hall sensors. *Intelligent Service Robotics*, 3(2) :99–113, 2010.
- [61] Chong, K. S. and Kleeman, L. Accurate odometry and error modelling for a mobile robot. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 2783–2788, 1997.
- [62] Codas, A. and Devy, M. and Lemaire, C. Robot localization algorithm using odometry and RFID technology. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 43(16) :569–574, 01 2010.
- [63] Colle, E. and Galerne, S. Mobile robot localization by multiangulation using set inversion. *Robotics and Autonomous Systems*, 61(1) :39–48, 2013.
- [64] Colle, E. and Galerne, S. A multihypothesis set approach for mobile robot localization using heterogeneous measurements provided by the Internet of Things. *Robotics and Autonomous Systems*, 96 :102–113, 06 2017.
- [65] Colle, E. and Galerne, S. A robust set approach for mobile robot localization in ambient environment. *Autonomous Robots*, 43(3) :557–573, 04 2019.
- [66] Cotera, P. and Velazquez, M. and Cruz, D. and Medina, L. and Bandala, M. Indoor Robot Positioning Using an Enhanced Trilateration Algorithm. *International Journal of Advanced Robotic Systems*, 13 :1–8, 06 2016.
- [67] M. Culleton, C. Mc Ginn, and K. Kelly. Framework for Assessing Robotic Dexterity within Flexible Manufacturing. *Journal of Intelligent and Robotic Systems*, 87(3-4) :507–529, 2017.
- [68] Di Paolo, E. A. and Hilizuka, I. How (not) to model autonomous behaviour. *Biosystems*, 91(2) :409 – 423, 2008. Modelling Autonomy.
- [69] Dijkstra, E. W. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1) :269–271, 1959.
- [70] Dreville, V. and Bonnifait, P. Robust positioning using relaxed constraint-propagation. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4843–4848, Oct 2010.

- [71] Drevelle, V. and Bonnifait, P. Localization Confidence Domains via Set Inversion on Short-Term Trajectory. *IEEE Transactions on Robotics*, 29(5) :1244–1256, Oct 2013.
- [72] B. Durand. *Proposition d'une architecture de contrôle adaptative pour la tolérance aux fautes*. Thèse de doctorat, Université de Montpellier, 2011.
- [73] Dąbkowski, M. and Skrzek, P. and Redlarski, G. Behavior Based Complete Coverage Task of Unknown Area by an Autonomous Mobile Robot SCORPION with Static Obstacles in Environment. *Solid State Phenomena*, 198 :73–78, 03 2013.
- [74] Eiben, A. E. and Smith, James E. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, 2003.
- [75] Espiau, B. *La Science au Présent 2001*. Encyclopédia Universalis, Nov 2000.
- [76] Esteves, J. S. and Carvalho, A. and Couto, C. Generalized geometric triangulation algorithm for mobile robot absolute self-localization. In *2003 IEEE International Symposium on Industrial Electronics*, volume 1, pages 346–351, 07 2003.
- [77] Faisal, A. and Yigitcanlar, T. and Kamruzzaman, M. and Currie, G. Understanding autonomous vehicles : A systematic literature review on capability, impact, planning and policy. *Journal of Transport and Land Use*, 12(1) :45–72, january 2019.
- [78] Fernández-Caramés and Carlos and Serrano and F Javier and Moreno and Vidal and Curto, Belén and Rodríguez-Aragón and J F and Alves, Raúl. A real-time indoor localization approach integrated with a Geographic Information System (GIS). *Robotics and Autonomous Systems*, 75 :475–489, 2016.
- [79] Filliat, D. Cartographie et localisation simultanée en robotique mobile. In *Techniques de l'ingénieur*. Techniques de l'ingénieur, 2014.
- [80] Filliat, D. and Meyer, J. A. Map-based navigation in mobile robots - I. A review of localisation strategies. *Journal of Cognitive Systems Research*, 4(4) :243–282, 2003.
- [81] Fischler, M. A. and Bolles, R. C. Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6) :381–395, jun 1981.
- [82] Ford, L. R. *Network Flow Theory*. RAND Corporation, 1956.
- [83] Forney, G. D. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3) :268–278, March 1973.
- [84] Fox, D. and Burgard, W. and Thrun, S. Markov Localization for Mobile Robots in Dynamic Environments. *Journal of Artificial Intelligence Research*, 11(1) :391–427, jul 1999.
- [85] Fu, G. and Zhang, J. and all. Precise Localization of Mobile Robots via Odometry and Wireless Sensor Network. *International Journal of Advanced Robotic Systems*, 10(4) :203, 2013.
- [86] Gai, S. and Jung, E. and Yi, B. Multi-group localization problem of service robots based on hybrid external localization algorithm with application to shopping mall environment. *Intelligent Service Robotics*, 9(3) :257–275, 2016.
- [87] Gallois, P. M. *Entreprise industrielle et typologie*. Technical report, Synthèse des travaux de la commission de pilotage et évaluation de la performance industrielle AFGI (in french), 1992.
- [88] Garrido-Jurado, S. and Muñoz-Salinas, R. and Madrid-Cuevas, R. J. and Marin-Jiménez, M. J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6) :2280 – 2292, 2014.

- [89] Ghallab, M. and Ingrand, F. An Overview of Problems and Approaches in Machine Intelligence. *Frontiers in Science and Engineering*, 6(1) :115–156, 2016.
- [90] Gning, A. and Bonnifait, P. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7) :1167 – 1175, 2006.
- [91] N. Gobillot, C. Lesire, and D. Doose. A Modeling Framework for Software Architecture Specification and Validation. In Davide Brugali, Jan F. Broenink, Torsten Kroeger, and Bruce A. MacDonald, editors, *Simulation, Modeling, and Programming for Autonomous Robots - 4th International Conference, SIMPAR 2014, Bergamo, Italy, October 20-23, 2014. Proceedings*, volume 8810 of *Lecture Notes in Computer Science*, pages 303–314. Springer, 2014.
- [92] Golledge, R. G. *Encyclopedia of Applied Psychology*, chapter Spatial cognition, page 443–452. Elsevier, 2004.
- [93] Gonzalez, R. and Rodriguez, F. and all. Combined visual odometry and visual compass for off-road mobile robots localization. *Robotica*, 30(6) :865–878, 2012.
- [94] Goodfellow, I. and Bengio, Y. and Courville, A. *Deep Learning*. MIT Press, 2016.
- [95] A. Grous. *Éléments d’analyse de la fiabilité et du contrôle de qualité : Statistiques appliquées par l’exemple*. Hermès - Lavoisier, 2013.
- [96] Grusser, O. J. *Spatially oriented behavior*, chapter Multimodal structure of the extrapersonal space, pages 327–352. Springer-Verlag, 1983.
- [97] R. Guyonneau. *Méthodes ensemblistes pour la localisation en robotique mobile*. Thèse de doctorat, Université de Nantes Angers Le Mans, 2013.
- [98] Guyonneau, R. and Lagrange, S. and Hardouin, L. A Visibility Information for Multi-Robot Localization. In *IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, pages 1426–1431, Tokyo, Japan, 2013.
- [99] Guyonneau, R. and Lagrange, S. and Hardouin, L. and Lucidarme, P. Guaranteed interval analysis localization for mobile robots. *Advanced Robotics*, 28(16) :1067–1077, Jul 2014.
- [100] Hart, P. E. and Nilsson, N. J. and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2) :100–107, 1968.
- [101] Hereau, A. and Godary-Dejean, K. and Guiochet, J. and Robert, C., and Claverie, T. and Crestani, D. Testing an Underwater Robot Executing Transect Missions in Mayotte. In *Toward Autonomous Robotic Systems, TAROS 2020*, page , 2020.
- [102] Humberto Herrero-Pérez, David and Alcaraz-Jimenez, Juan Jose and Martínez-Barberá. Mobile Robot Localization Using Fuzzy Segments. *International Journal of Advanced Robotic Systems*, 10(12) :406, 2013.
- [103] Ho, D. K. *QoS-Aware Resource and Energy Management for Autonomous Mobile Robotic Systems*. Thèse de doctorat, Université de Côte d’Azur, 2020.
- [104] Ho, D. K. and Chehida, K. B. and Miramond, B. and Auguin, M. Learning-Based Adaptive Management of QoS and Energy for Mobile Robotic Missions. *International Journal of Semantic Computing*, 13 :513–539, 12 2019.
- [105] IEEE 1873-2015. IEEE Standard for Robot Map Data Representation for Navigation. Technical report, IEEE, Oct 2015.
- [106] IEEE Robotics and Automation Society. Performance Evaluation and Benchmarking of Robotics and Automation Systems. <https://www.ieee-ras.org/performance-evaluation>.

- [107] IEEE Standards Association. IEEE Standard for Robot Map Data Representation for Navigation. Technical report, (IEEE Robotics and Automation Society), 2015.
- [108] F. Ingrand. LAAS architecture : Open robot. In *1st National Conference on Control Architectures of Robots*, 2006.
- [109] Ingrand, F. and Ghallab, M. Deliberation for autonomous robots : A survey. *Artificial Intelligence*, 247 :10–44, June 2017.
- [110] Irie, K. and Sugiyama, M. and Tomono, M. Dependence maximization localization : a novel approach to 2D street-map-based robot localization. *Advanced Robotics*, 30(22) :1431–1445, 2016.
- [111] Ishikawa, K. and Lu, D. J. *What is total quality control ? : the Japanese way*. Prentice-Hall Englewood Cliffs, 1985.
- [112] Jaiem, L. *Contribution à l'autonomie des robots : Vers la garantie de performance en robotique mobile par la gestion des ressources matérielles et logicielles*. Thèse de doctorat, Université de Montpellier, 2016.
- [113] Jaiem, L. and Lapierre, L. and Godary-Dejean, K. and Crestani, D. A Step Toward Mobile Robots Autonomy : Energy Estimation Models. In *Toward Autonomous Robotic Systems, TAROS 2016*, pages 177–188, 06 2016.
- [114] Jaiem, L. and Lapierre, L. and Godary-Dejean, K. and Crestani, D. Fault tolerant autonomous robots using mission performance guided resources allocation. In *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 45–50, Sep. 2016.
- [115] Jaiem, L. and Lapierre, L. and Godary-Dejean, K. and Crestani, D. Toward Performance Guarantee for Autonomous Mobile Robotic Mission : An Approach for Hardware and Software Resources Management. In *Toward Autonomous Robotic Systems, TAROS 2016*, pages 189–195, 06 2016.
- [116] Jaulin, L. A Nonlinear Set-membership Approach for the Localization and Map Building of an Underwater Robot using Interval Constraint Propagation. *IEEE Transactions on Robotics*, 25(1) :88–98, 2009.
- [117] Jaulin, L. and Chabert, G. QUIMPER : un langage de programmation pour le calcul ensembliste ; Application à l'automatique. In *CIFA*, Lille, France, Feb 2008.
- [118] Jaulin, L. and Kieffer, M. and Didrit, O. and Walter, E. *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer London Ltd, Aug 2001.
- [119] Jaulin, L. and Walter, E. Set inversion via intervalanalysis for nonlinear bounded-error estimation. *Automatica*, 29(4) :053–1064, 1993.
- [120] Jensfelt, P. and Kristensen, S. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transaction on Robotics and Automation*, 17 :748–760, 1999.
- [121] Julier, S. J. and Uhlmann, J. K. A new extension of the kalman filter to nonlinear systems. In *The Proceeding of Aerosense : The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Control*, 1997.
- [122] Jung, J. and Li, J. and all. Localization of AUVs using visual information of underwater structures and artificial landmarks. *Intelligent Service Robotics*, 10(1) :67–76, 2017.
- [123] Kalman, R. E. A New Approach to Linear Filtering And Prediction Problems. *ASME Journal of Basic Engineering*, 82(1) :35–45, 1960.
- [124] Kant, I. *Kant : Philosophical Correspondance, 1759, 1799*. University of Chicago Press, 1967.



- [125] Kejriwal, N. and Kumar, S. and Shibata, T. High performance loop closure detection using bag of word pairs. *Robotics and Autonomous Systems*, 77 :55–65, 3 2016.
- [126] Kelly, A. Linearized Error Propagation in Odometry. *International Journal of Robotics Research*, 23 :179–218, 2004.
- [127] Kenneth, S. Review of Stochastic Processes and Filtering Theory - Andrew H. Jazwinski. *IEEE Transactions on Automatic Control*, 17 :752– 753, 11 1972.
- [128] Kieffer, M. and Jaulin, L. and Walter, E. and Meizel, D. . Guaranteed Mobile Robot Tracking Using Interval Analysis. In *Proceedings of the MISC'99 Workshop on Applications of Interval Analysis to Systems and Control*, pages 347–359, Girona, Spain, 1999.
- [129] Kim, J. and Park, J. and Chung, W. Self-Diagnosis of Localization Status for Autonomous Mobile Robots. *Sensors*, 18(9), 2018.
- [130] Kim, T. and Kim, J. and Choi, H. Mobile robot navigation using grid line patterns via probabilistic measurement modeling. *Intelligent Service Robotics*, 9(2) :141–151, 2016.
- [131] Kirianaki, N. V. and Mizuochi, M. and Nishi, H. and Ohnishi, N. O. A velocity measurement method for acceleration control. In *Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society - IECON 2005*, volume 70, pages 1943–1948, 12 2005.
- [132] Kneip, L. and Tache, F. and Caprari, G. and Siegwart, R. Characterization of the compact Hokuyo URG-04LX 2D laser range scanner. In *2009 IEEE International Conference on Robotics and Automation*, pages 1447–1454, 05 2009.
- [133] Knuth, J. and Barooah, P. Error growth in position estimation from noisy relative pose measurements. *Robotics and Autonomous Systems*, 61 :229–244, 03 2013.
- [134] Koubaa, A. *Robot Operating System (ROS) : The Complete Reference (Volume 2)*. Springer Publishing Company, Incorporated, 1st edition, 2017.
- [135] Kragten, G. A. and Kool, A. C. and Herder, J. L. Ability to hold grasped objects by underactuated hands : Performance prediction and experiments. In *2009 IEEE International Conference on Robotics and Automation*, pages 2493–2498, May 2009.
- [136] Krajnik, T. and Nitsche, M. and Faigl, J. and Duckett, T. and Mejail, M. and Preucil, L. External Localization System for Mobile Robotics. In *2013 16th International Conference on Advanced Robotics, ICAR 2013*, pages 1–6, 12 2013.
- [137] Kreinar, E. J. and Quinn, R. D. . Odometry error estimation for a differential drive robot snowplow. In *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 1122–1129, 2014.
- [138] Kubelka, V. and Oswald, L. and Pomerleau, F. and Colas, F. and Svoboda, T. and Reinstein, M. Robust data fusion of multimodal sensory information for mobile robots. *Journal of Field Robotics*, 32(4) :447–473, 2015.
- [139] Kummerle, R. and al. On Measuring the Accuracy of SLAM Algorithms. *Autonomous Robots*, 27(4) :387–407, nov 2009.
- [140] Lambert, A. and Gruyer, D. and Vincke, B. and Seigne, E. Consistent Outdoor Vehicle Localization by Bounded-Error State Estimation. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1211–1216, 2009.
- [141] Lambert, P. and Lapierre, L. and Crestani, D. An approach for Fault Tolerant and Performance Guarantee Autonomous Robotic Mission. In *13th NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2019*, pages 87–94, 07 2019.
- [142] Landau, B. *Encyclopedia of the Human Brain*, chapter Spatial cognition, page 395–418. Elsevier, 2002.

- [143] Lapiere, L. and Zapata, R. A guaranteed obstacle avoidance guidance system : The safe maneuvering zone. *Autonomous Robots*, pages 177–187, 2012.
- [144] Lee, S. and Tewolde, G. and Lim, J. and Kwon, J. QR-code based Localization for Indoor Mobile Robot with validation using a 3D optical tracking instrument. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 965–970, 2015.
- [145] Levitt, T. S. and Lawton, D. T. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44(3) :305–360, 1990.
- [146] Li, G. and Fu, Y. and Zhang, F. and Qu, J. and Yuan, R. An enhanced pose tracking method using progressive scan matching. *Industrial Robot : the international journal of robotics research and application*, 46(2) :235–246, jan 2019.
- [147] Liu, H. and Jiang, R. and Hu, W. and Wang, S. Navigational Drift Analysis for Visual Odometry. *Computing and Informatics*, 33 :685–706, 2014.
- [148] Llarena, A. and Savage, J. and Kuri, A. and Escalante-Ramírez, B. Odometry-Based Viterbi Localization with Artificial Neural Networks and Laser Range Finders for Mobile Robots. *Journal of Intelligent and Robotic Systems - JIRS*, 66 :75–109, april 2012.
- [149] Loevsky, I. and Shimshoni, I. Reliable and efficient landmark-based localization for mobile robots. *Robotics and Autonomous Systems*, 58(5) :520–528, 2010.
- [150] S. Louis. *Système robotisé semi-autonome pour l’observation des espèces marines*. Thèse de doctorat, Université de Montpellier, 2018.
- [151] Luck, M. and D’Inverno, M. and Munroe, S. *Agent Autonomy. Multiagent Systems, Artificial Societies, and Simulated Organizations*, chapter Autonomy : Variable and Generative. International Book Serie, Springer, Boston, MA, 2003.
- [152] Luo, Y. and Tao, J. and all. A novel localization approach for underwater welding vehicles in spent fuel pools via attitude heading reference system and altimeters. *International Journal of Advanced Robotic Systems*, 16(2) :1729881419830540, 2019.
- [153] Lygouras, E. and al. Unsupervised Human Detection with an Embedded Vision System on a Fully Autonomous UAV for Search and Rescue Operations. *Sensors*, 19(16) :35–42, 2019.
- [154] Lyons, D. and Arkin, R. and al. A Software Tool for the Design of Critical Robot Missions with Performance Guarantees. *Procedia Computer Science*, 16 :888–897, 12 2013.
- [155] Lyons, D. and Arkin, R. and al. Performance Verification for Behavior-Based Robot Missions. *IEEE Transactions on Robotics*, 31 :1–36, 06 2015.
- [156] Lyons, D. and Arkin, R. and Nirmal, P. and Jiang, S. Designing autonomous robot missions with performance guarantees. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2583–2590, 10 2012.
- [157] Marion, A. and al. *Diagnostic de la performance d’entreprise, Concepts et Méthodes*. Dunod, 2012.
- [158] Martin-Bennefous, C. and Courtois, A. *Indicateurs de performance*. Hermès Science, 2001.
- [159] Martinelli, A. Modeling and Estimating the Odometry Error of a Mobile Robot. *IFAC Proceedings Volumes*, 34(6) :407–412, 2001. 5th IFAC Symposium on Nonlinear Control Systems 2001, St Petersburg, Russia, 4-6 July 2001.
- [160] Masse, J. T. *Capture de mouvements humains par capteurs RGB-D*. Thèse de doctorat, Université Toulouse III - Paul Sabatie, 2015.

- [161] McCann, E. and Medvedev, M. and Brooks, D. J. and Saenko, K. Off the grid : Self-contained landmarks for improved indoor probabilistic localization. In *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6, 2013.
- [162] Merriaux, P.e and Dupuis, Y. and Boutteau, R. and Vasseur, P. and Savatier, X. A Study of Vicon System Positioning Performance. *Sensors*, 17(7), Jul 2017.
- [163] Meyer, D. Glider Technology for Ocean Observations : A Review. *Ocean Science Discussions*, pages 1–26, 2016.
- [164] Mi, J. and Yasutake, T. Performance analysis of mobile robot self-localization based on different configurations of RFID system. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 07 2015.
- [165] Mikulová, Z. and Duchoň, F. and all. Localization of mobile robot using visual system. *International Journal of Advanced Robotic Systems*, 14(5) :1729881417736085, 2017.
- [166] MobileRobots. Pioneer 3 Operations Manual. Technical report, MobileRobots Inc., 2006.
- [167] Montemerlo, M. and Roy, N. and Thrun, S. CARMEN - Robot Navigation Toolkit. <http://carmen.sourceforge.net/>, Jul 2012.
- [168] Morbidi, F. *Localisation et Navigation de robots*. Cours M2 - Université de Picardie, 2015.
- [169] Murphy, R. R. *Introduction to A.I. Robotics*. Cambridge, MA : The MIT Press, 2000.
- [170] Nazemzadeh, P. and Fontanelli, D. and Macii, D. and Palopoli, L. Indoor Localization of Mobile Robots Through QR Code Detection and Dead Reckoning Data Fusion. *IEEE/ASME Transactions on Mechatronics*, 22(6) :2588–2599, 2017.
- [171] Neuland, R. and Nicola, J. and Maffei, R. and Jaulin, L. and Prestes, E. and Kolberg, M. Hybridization of Monte Carlo and Set-membership Methods for the Global Localization of Underwater Robots. In *IEEE/RJS International Conference on Intelligent Robots and Systems (IROS) 2014*, pages 199–204, 2014.
- [172] Nister, D. and Naroditsky, O. and Bergen, J. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1) :3–20, 2006.
- [173] Okubo, Y. and Ye, C. and Borenstein, J. Characterization of the Hokuyo URG-04LX Laser Rangefinder for Mobile Robot Obstacle Negotiation. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 7332, pages 279 – 288, 05 2009.
- [174] Olson, C. and Matthies, L. and Schoppers, M. and Maimone, M. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43 :215–229, 2003.
- [175] Olson, C. F. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1) :55–66, 2000.
- [176] Open-CV team. Open Source Computer Vision Library. <https://opencv.org/>, 2020.
- [177] Pagliari, D. and Pinto, L. Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. *Sensors*, 15 :27569–27589, 2015.
- [178] Parasuraman, R. and Sheridan, T. B. and Wickens, C. D. A Model for Types and Levels of Human Interaction with Automation. *Transaction System Man and Cybernetic - Part A*, 30(3) :286–297, may 2000.
- [179] Park, J. J. *Graceful Navigation for Mobile Robots in Dynamic and Uncertain Environments*. Thèse de doctorat, Université du Michigan, 2016.
- [180] Park, J. J. and Kuipers, B. Stability Guaranteed MPC for Mobile Robot Navigation. In *ICRA 2015 Workshop on Optimal Robot Motion Planning*, 2015.

- [181] Passama, R. ContrACT : une méthodologie de conception et de développement d’architectures de contrôle de robots. Technical report, LIRMM - Rapport interne RR-10025, 2010.
- [182] Passama, R. and Andreu, D. ContrACT : a software environment for developing control architecture. In *CAR : Control Architectures of Robots*, Grenoble, France, May 2011. INRIA Grenoble Rhône-Alpes.
- [183] Passama, R. and Andreu, D. and Godary-Dejean, K. and Crestani, D. Architectures de contrôle pour la robotique - Approches et tendances. In *Techniques de l’ingénieur*. Techniques de l’ingénieur, Sep 2014.
- [184] Penedo, C. and Pavão, J. and Lima, P. and Ribeiro, M. I. Markov Localization in the Robocop Simulation League. In *Proc. ROBOTICA2003 - 3rd Portuguese Robotics Festival, Lisbon, Portugal*, 2003.
- [185] Piasco, N. and Sidibé, D. and Demonceaux, C.c and Gouet-Brunet, V. A survey on Visual-Based Localization : On the benefit of heterogeneous data. *Pattern Recognition*, 74 :90–109, Feb 2018.
- [186] M. Piaseck. Global localization for mobile robots by multiple hypothesis tracking. *Robotics and Autonomous Systems*, 16(1) :93–104, 1995.
- [187] Pondrúška, P. and Gurău, C. and Marchegiani, L. and al. Scheduled perception for energy-efficient path following. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4799–4806, May 2015.
- [188] Porta, J. and Verbeek, J. and Krose, B. Active appearance-based robot localization using stereo vision. *Autonomous Robots*, 18(1) :59–80, 2005.
- [189] Quigley, M. and Conley, K. and Gerkey, B. and Faust, J. and Foote, T. and Leibs, J. and Wheeler, R. and Ng, A. ROS : an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, volume 3, 01 2009.
- [190] ANSI/RIA R15.06. Industrial Robots and Robot Systems - Safety Requirements. Technical report, American National Standards Institute, 1992.
- [191] ANSI/RIA R15.05-2-1992 (R1999). Industrial Robots and Robot Systems - Path-Related and Dynamic Performance Characteristics - Evaluation. Technical report, American National Standards Institute, 1992.
- [192] Rached, D. and Hatab, A. A. Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies : A Unified Framework. *Advances in Robotics and Automation*, 02(2), 01 2013.
- [193] Rano, I. and Minguez, J. Steps towards the automatic evaluation of robot obstacle avoidance algorithms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6, 2006.
- [194] Simon Rohou, Peter Franek, Clément Aubry, and Luc Jaulin. Proving the existence of loops in robot trajectories. *The International Journal of Robotics Research*, 37(12) :1500–1516, October 2018.
- [195] Rohou, S. *Reliable robot localization : a constraint programming approach over dynamical systems*. Thèse de doctorat, Université de Bretagne Occidentale, 2017.
- [196] Rohou, S. and Jaulin, L. and Mihaylova, L. and Le Bars, F. and Veres, S. M. Guaranteed computation of robot trajectories. *Robotics and Autonomous Systems*, 93 :76 – 84, July 2017.
- [197] Rohrmüller, F. and al. System interdependence analysis for autonomous mobile robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 696–701, Oct 2009.

- [198] Rohrmüller, F. and al. Interconnected performance optimization in complex robotic systems. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4113–4118, 10 2010.
- [199] Romero-Ramirez, F. and Munoz-Salinas, R. and Medina-Carnicer, R. Speeded Up Detection of Squared Fiducial Markers. *Image and Vision Computing*, 76 :38–47, 06 2018.
- [200] Roumeliotis, S. I. and Bekey, G. A. Bayesian estimation and Kalman filtering : a unified framework for mobile robot localization. In *IEEE International Conference on Robotics and Automation, Millenium conference*, volume 3, pages 2985–2992, 2000.
- [201] Rudolph, A. Quantification and Estimation of Differential Odometry Errors in Mobile Robotics with Redundant Sensor Information. *International Journal of Robotics Research*, 22 :117–128, 02 2003.
- [202] Sarbolandi, H. and Lefloch, D. and Kolb, A. Kinect range sensing : Structured-light versus Time-of-Flight Kinect. *Journal of Computer Vision and Image Understanding*, 139 :1–20, 2015.
- [203] S. Segvic, A. Remazeilles, A. Diosi, and François Chaumette. Mapping and localization framework for scalable appearance-based navigation. *Computer Vision and Image Understanding*, 133 :172–187, 2009.
- [204] Siegwart, R. and Nourbakhsh, I. R. *Introduction to Autonomous Mobile Robots*. Bradford Company, USA, 2004.
- [205] Silvain, L. and Godary-Dejean, K. and Lapierre, L. and al. Formal Method for Mission Controller Generation of a Mobile Robot. In *TAROS : Towards Autonomous Robotic Systems*, pages 586–600, University of Surrey, Guildford, United Kingdom, 2017.
- [206] Simandl, M. and Straka, O. Sampling Densities of Particle Filter : A Survey and Comparison. In *2007 American Control Conference*, pages 4437 – 4442, 08 2007.
- [207] Simon, H. A. *The Sciences of the Artificial*. Cambridge, MA : The MIT Press, 1996.
- [208] Skinner, B. F. *Reflection on Behaviorism and Society*. Englewood Cliffs, N.J. : Prentice-Hall, 1978.
- [209] Sobreira, H., Costa, C.M., Sousa, I. et al. Map-Matching Algorithms for Robot Self-Localization : A Comparison Between Perfect Match, Iterative Closest Point and Normal Distributions Transform. *Journal of Intelligent and Robotic Systems*, 93 :533–546, 03 2019.
- [210] Soetens, P. and Herman, B. Realtime hybrid task-based control for robots and machine tools. In *International Conference on Robotics and Automation (ICRA)*, pages 259–264, 2005.
- [211] Straka, O. and Simandl, M. A Survey of Sample Size Adaptation Techniques for Particle Filters. In *15th IFAC Symposium on System Identification*, volume 42, pages 1358 – 1363, 2009.
- [212] Tahon, C. *Évaluation des performances des systèmes de production*. Lavoisier, 2003.
- [213] Tahon, C. and Frein, Y. Document de synthèse du G.R.P. Thème 4 : Évaluation des performances. Technical report, Groupement de Recherche en Productique (in french), 2000.
- [214] Takuma, N. and Genya, I. Gyro-based odometry associated with steering characteristics for wheeled mobile robot in rough terrain. *Advanced Robotics*, 30(23) :1495–1508, 2016.
- [215] Tarin Sauer, C. and Brugger, H. and Hofer, E. P. and Tibken, B. Odometry error correction by sensor fusion for autonomous mobile robot navigation. In *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference*, volume 3, pages 1654–1658, 2001.

- [216] Thorndyke, P. W. and Hayes-Roth, B. Differences in spatial knowledge acquired from maps and navigation. *Cognitive psychology*, 14(4) :560—589, October 1982.
- [217] Thrun, S. Particle Filters in Robotics. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, page 511–518, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [218] Thrun, S. Robotic Mapping : A Survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [219] Thrun S. Toward a Framework for Human-Robot Interaction. *Human-Computer Interaction*, 19(1-2) :9–24, 2004.
- [220] Thrun S. The Robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9) :661–692, 2006.
- [221] Thrun, S. and Burgard, W. and Fox, D. *Probabilistic Robotics*. MIT Press, 2005.
- [222] TOTAL. ARGOS Challenge : Autonomous Robots for Gas and Oil Sites, 2014.
- [223] Trojnecki, M. and Dąbek, P. Mechanical Properties of Modern Wheeled Mobile Robots. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 13 :3–13, 12 2019.
- [224] Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2) :260–269, 1967.
- [225] Volpe, R. Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype. *The International Journal of Robotics Research*, 18(7) :669–683, 1999.
- [226] Volpe, R. and Nesnas, I. and Estlin, T. and Mutz, D. and Petras, R. and Das, H. The CLARATY architecture for robotic autonomy. In *IEEE Aerospace Conference*, volume 1, pages 121–132, 2001.
- [227] Wang, C. M. Location estimation and uncertainty analysis for mobile robots. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1231–1235, 1988.
- [228] Weiss, G. and Puttkamer, E. A map based on laserscans without geometric interpretation. *Intelligent Autonomous Systems*, 4 :403–407, 01 1995.
- [229] Wong, S. C. and Middleton, L. and Mac Donald, B. A. Performance Metrics for Robot Coverage Tasks. In *Australian Conference on Robotics and Automation (ACRA)*, pages 7–12, 2002.
- [230] Wooldridge, M. and Jennings, N. R. Intelligent agents : theory and practice. *The Knowledge Engineering Review*, 10(2) :115–152, 1995.
- [231] Woolley, B. and Peterson, G. and Kresge, J. Real-time behavior-based robot control. *Autonomous Robots*, 30 :233–242, 04 2011.
- [232] Xiong, X. and Li, Y. and Qin, H. Structural Dynamic Performance Evaluation of Industrial Robots Based on Vibration Tests. In *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 302–306, Dec 2018.
- [233] Yilmaz, A. and Temeltas, H. Self-adaptive Monte Carlo method for indoor localization of smart AGVs using LIDAR data. *Robotics and Autonomous Systems*, 122 :103–285, 2019.
- [234] Yoon-Gu, K. and Jinung, A. and Ki-Dong, L. Localization of Mobile Robot Based on Fusion of Artificial Landmark and RF TDOA Distance under Indoor Sensor Network. *International Journal of Advanced Robotic Systems*, 8(4) :52, 2011.
- [235] Yun, X. and Yamamoto, Y. Internal dynamics of a wheeled mobile robot. In *IEEE/RJS International Conference on Intelligent Robots and Systems (IROS) 1993*, pages 1288–1294, 1993.

- [236] Zang, C. and Na, J. and Whang, B. and Gao, G. Prescribed performance control for robotic systems with unknown dynamics. In *Chinese Intelligent Systems Conference*, pages 467–476, 2018.
- [237] Zapata, R. and Jaiem, L. and Zhang, L. and Crestani, D. SL2 : a novel algorithm for mobile robot localization. Technical report, LIRMM/CNRS - Equipe EXPLORE, 2020.
- [238] Zhang, F. and Bertozzi, A. L. and Elamvazhuthi, K. and Berman, S. Performance Bounds on Spatial Coverage Tasks by Stochastic Robotic Swarms. *IEEE Transactions on Automatic Control*, 63(6) :1563–1578, June 2018.
- [239] Zhang, H. and Zhang, C. and Yang, W. and Chen, C. Localization and navigation using QR code for mobile robot in indoor environment. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2501–2506, 2015.
- [240] Zhang, L. and Zapata, R. and Lépinay, P. Self-adaptive monte carlo localization for mobile robots using range finders. *Robotica*, 30(2) :229–244, 2012.
- [241] Zhang, X. and Zhu, S. and all. Hybrid visual natural landmark–based localization for indoor mobile robots. *International Journal of Advanced Robotic Systems*, 15(6) :1729881418810143, 2018.
- [242] Zhao, C. and Yang, J. and Wang, C. Monocular odometry in country roads based on phase-derived optical flow and 4-DOF ego-motion model. *Industrial Robot : An International Journal*, 38(5) :509–520, jan 2011.
- [243] Zheltoukhov, A. A. and Stankevich, L. A. A survey of control architectures for autonomous mobile robots. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 1094–1099, Feb 2017.
- [244] Zhu, J. and Zheng, N. and Yuan, Z. An Improved Technique for Robot Global Localization in Indoor Environments. *International Journal of Advanced Robotic Systems*, 8 :21–28, 03 2011.
- [245] Zieba, S. and Polet, P. and Vanderhaegen, F. Using adjustable autonomy and human–machine cooperation to make a human–machine system resilient – Application to a ground robotic system. *Information Sciences*, 181 :379–397, 02 2011.

## Contribution à l'autonomie des robots : Vers des mission à garantie de performance incluant la localisation en environnement intérieur connu

### Résumé :

La garantie de performance de missions robotiques autonomes, bien qu'indispensable, est encore très rarement recherchée de nos jours.

Le travail présenté dans ce manuscrit fait suite à celui proposé dans la thèse de L. Jaiem qui adressait de façon quantitative les dimensions sécuritaire, durée et énergie de la performance de missions planifiées en environnement connu. La méthodologie PANORAMA(SED) (Performance and AutoNOmy using Resources Allocation MAnagement) proposée s'appuyait sur une gestion dynamique et éclairée des ressources matérielles et logicielles disponibles pour parvenir à garantir, en temps réel, les objectifs de performance fixés. La mission était alors décomposée en une succession d'activités indépendantes à contraintes constantes où les ressources à employer étaient identifiées et paramétrées. La dimension localisation n'y était considérée que de façon qualitative et la logique d'utilisation de l'énergie s'appuyait sur une logique de minimisation de la marge énergétique.

Dans ce document, l'intégration de la prise en compte qualitative de la dimension localisation de la performance au travers de l'incertitude associée repose sur la construction de modèles prédictifs d'incertitude de pose s'appuyant sur une démarche ensembliste, largement établis de façon empirique. La méthodologie proposée est déployée sur des méthodes locales couplant l'odométrie avec un recalage extéroceptif faisant appel à des capteurs Kinect et à des marqueurs Aruco. Elle est aussi mise en œuvre sur une nouvelle méthode de localisation globale basée sur les apparences, LZA, s'appuyant sur la corrélation entre la signature extéroceptive courante du robot et celles, préalablement calculées, au sein d'une grille couvrant l'environnement.

La prise en compte qualitative de la performance d'incertitude de localisation fait apparaître qu'une corrélation peut exister entre 2 activités successives en raison des délais d'initialisation nécessaires à l'opérationnalisation d'une nouvelle méthode de localisation. Cette nouvelle représentation des données couplée aux modèles d'incertitude de pose prédictifs permet d'utiliser l'algorithme de Viterbi pour sélectionner les méthodes de localisation (ressources matérielles et logicielles) à mettre en œuvre pour satisfaire les contraintes de pose définies. Mais surtout il permet d'adresser la performance énergétique dans une logique de maximisation de la marge associée. Une nouvelle méthodologie PANORAMA(SLED) enrichissant PANORAMA(SED) a donc été proposée.

La méthodologie PANORAMA(SLED) est déployée expérimentalement sur une mission de Patrouille exécutée en environnement intérieur présentant un fort aliasing perceptuel. Elle a montré sa capacité à garantir les objectifs de performance Sécurité – Énergie – Localisation – Durée imposés. Les échecs observés ont pour principale origine la non prise en compte de l'incertitude de localisation lors de la commutation interactivités.

Même si PANORAMA(SLED) présente encore quelques limitations elle devrait, dans les prochaines années, être étendue aux missions d'Exploration, en permettant une optimisation objective de la phase de découverte, tout en assurant la phase de récupération du robot autonome.

**Mots clef :** Robotique mobile - Autonomie - Performance - Garantie - Localisation



## Contribution to the autonomy of robots : Towards missions with performance guarantee including localization in known indoor environment

### Abstract :

The performance guarantee of autonomous robotic missions is yet rarely addressed.

This work follows the work proposed in previous PhD which quantitatively deals with the safety, duration and energy dimensions of the performance of planned missions in known environments. The PANORAMA (SED) (Performance and AutoNOmy using Resources Allocation Management) methodology proposed manages dynamically the available hardware and software resources to guarantee, in real time, the performance objectives. The mission is split into a succession of independent activities with constant constraints where the resources to be employed are identified and configured. The localization dimension is only considered qualitatively and energy issue is addressed by minimizing the energy margin.

In this document, the integration of the qualitative consideration of the localization dimension of performance is based on the construction of predictive uncertainty models of pose, largely established empirically, based on an interval approach. The proposed methodology is deployed on local localization methods using odometry with an exteroceptive resetting using Kinect sensors and Aruco markers. It also implements on a new global localization method based on appearances called LZA. It is based on the correlation between the current exteroceptive signature of the robot and those, previously calculated, within a grid covering the environment.

Taking into account the localization performance qualitatively shows that a correlation can exist between 2 successive activities due to the initialization times required for the operationalization of a new localization method. This new representation of the data coupled with predictive pose uncertainty models makes possible to use the Viterbi algorithm to select the localization methods (hardware and software resources) to be implemented to satisfy the defined pose constraints. It makes also possible to address the energy performance to maximize the associated margin. A new PANORAMA (SLED) methodology enriching PANORAMA (SED) is therefore proposed.

The PANORAMA (SLED) methodology is deployed experimentally on a Patrol mission executed in an indoor environment with strong perceptual aliasing. It has shown its ability to guarantee the performance objectives Security - Energy - Location - Duration. The main shortcoming comes from the fact that presently we do not consider the localization uncertainty when switching between activities.

Even if PANORAMA (SLED) still has some limitations, it should, in the coming years, be extended to Exploration missions, allowing objective optimization of the discovery phase, while ensuring the recovery phase of the autonomous robot.

**Key words :** Mobile robotics - Autonomy - Performance - Guarantee - Localization