



HAL
open science

De l'utilisation de l'algèbre différentielle pour la localisation et la navigation de robots mobiles autonomes

Hugues Sert

► **To cite this version:**

Hugues Sert. De l'utilisation de l'algèbre différentielle pour la localisation et la navigation de robots mobiles autonomes. Autre. Ecole Centrale de Lille, 2013. Français. NNT : 2013ECLI0002 . tel-00862870

HAL Id: tel-00862870

<https://theses.hal.science/tel-00862870>

Submitted on 17 Sep 2013

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.

Numéro d'ordre : 215
ÉCOLE CENTRALE DE LILLE

THÈSE
Présentée en vue
d'obtenir le grade de
DOCTEUR
en
Automatique, Génie Informatique, Traitement du Signal et Image

par
HUGUES SERT

DOCTORAT DÉLIVRÉ PAR L'ÉCOLE CENTRALE DE LILLE

DE L'UTILISATION DE L'ALGÈBRE
DIFFÉRENTIELLE
POUR LA LOCALISATION ET LA
NAVIGATION
DE ROBOTS MOBILES AUTONOMES

Soutenue le 11 janvier 2013 devant le jury d'examen :

Mr Michel Fliess	Ecole Polytechnique	Président du Jury
Mr Philippe Martin	Ecole des Mines de Paris	Rapporteur
Mr Philippe Bonnifait	Université Technologique de Compiègne	Rapporteur
Mr François Chaumette	INRIA Rennes	Rapporteur
Mr Patrick Rives	INRIA Sophia Antipolis	Membre du jury
Mr Gérard Lefranc	Thales	Membre du jury
Mr David Lemaitre	EOS Innovation	Membre du jury
Mr Wilfrid Perruquetti	École Centrale de Lille	Directeur de thèse
Mme Annemarie Kokosy	ISEN Lille	Co-encadrant de thèse

Thèse préparée dans le laboratoire LAGIS - SyNeR, INRIA NON-A
École doctorale SPI 072, École Centrale de Lille
PRES Université Lille Nord-de-France

À Marina et Margaux

REMERCIEMENTS

Lille, le 11 janvier 2013

Le travail que nous présentons dans ce mémoire a été effectué à l'ISEN Lille et à l'INRIA Lille sous la direction de Monsieur Wilfrid Perruquetti, Professeur à l'École Centrale de Lille et membre du projet INRIA NON-A et Madame Annemarie Kökösy, Enseignant chercheur à l'ISEN Lille. Je tiens à remercier très vivement Monsieur Wilfrid Perruquetti et Madame Annemarie Kökösy pour m'avoir accompagné durant ces trois années de thèse. Ils m'ont fait découvrir le monde de la robotique et m'ont, par leurs précieux conseils et leur disponibilité, permis d'apporter ma modeste pierre à l'édifice de la robotique mobile dans les meilleures conditions. Je tiens à leur exprimer toute mon amitié et ma reconnaissance.

Toute ma reconnaissance va aussi aux rapporteurs et aux membres du jury pour l'intérêt qu'ils ont montré pour mes travaux.

Ce travail a été soutenu financièrement par la Mission Insertion Thales, le FEDER et le programme Interreg IVA 2 Mers (projet FR-06-020 SYSIASS), l'Université Catholique de Lille et EOS Innovation.

Un grand merci à la Mission Insertion de Thales et à Monsieur Lefranc, son directeur, d'avoir cru en ce travail. Je remercie l'entreprise EOS Innovation pour la confiance qu'elle m'accorde en me donnant l'occasion aujourd'hui d'appliquer à une nouvelle problématique les différentes techniques développées pendant ma thèse, lui donnant une dimension industrielle. Cette expérience justifie à elle seule les longues heures de recherches nécessaires à l'écriture de ce manuscrit. Je remercie aussi les membres du projet Interreg Sysiass, la fondation Garches, le Groupe Hospitalier de l'Institut Catholique de Lille et les centres de rééducation Marc Sautet de Villeneuve d'Ascq et Jacques Calvé de Berck-sur-Mer, l'APF et son Réseau Nouvelles Technologies pour tous les échanges scientifiques et techniques que nous avons pu avoir pendant les deux dernières années de ma thèse, notamment sur la problématique du handicap.

Tout naturellement, j'exprime ma gratitude aux équipes SMART de l'ISEN Lille, NON-A de l'INRIA et SyNeR du Lagis, pour tout ce qu'elles m'ont offert. Un grand merci à tous pour leur présence et leurs conseils avisés tout au long de ce travail.

Un immense merci à l'ISEN et à tous mes collègues grâce auxquels ma thèse s'est déroulée dans des conditions idéales, que ce soit en terme d'épanouissement professionnel que d'esprit d'équipe.

Merci tout particulièrement à François et Xin, collègues de bureau attentifs et disponibles. Nous avons toujours pu discuter des divers aspects de la robotique

avec des approches liées à nos différentes sensibilités. Un grand merci à Jorge pour son aide et son amitié ainsi qu'à Dominique et Olivier pour tous les bons moments passés ensemble.

Toute ma reconnaissance va à Guy pour ses relectures avisées sans lesquelles ce manuscrit serait bien plus pénible à lire.

Un grand merci à Marina de m'avoir soutenu durant ces trois années malgré les incertitudes sur le futur liées à ce choix. J'embrasse tendrement Margaux, qui même si elle n'a pas suivi toute l'histoire, en a quand même vécu la fin. Je ne peux finir ces remerciements sans évoquer ma famille qui m'a toujours fourni le cadre idéal pour vivre mes passions et m'a soutenu dans mes choix sans me juger, et ce même si je me retrouve encore étudiant à 26 ans.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	7
LISTE DES FIGURES	11
NOTATIONS	13
1 INTRODUCTION	15
1.1 ROBOTS ET ROBOTIQUE DE SERVICE	17
1.1.1 Un robot, trois visions françaises différentes	17
1.1.2 Définition d'un robot	17
1.1.3 Robotique de service : un secteur en pleine croissance	18
1.1.4 Périmètres d'application	18
1.2 LA ROBOTIQUE MOBILE AUTONOME	18
1.2.1 Un premier exemple de robot autonome	18
1.2.2 Architecture d'un robot mobile autonome	20
1.3 LES CHALLENGES SCIENTIFIQUES DE CE TRAVAIL	21
1.4 LOCALISATION DE ROBOTS MOBILES	22
1.4.1 Vidéo : détection des points d'intérêt	23
1.4.2 Correspondance entre observations réelles et attendues	23
1.4.3 Bref état de l'art sur la localisation	24
1.4.4 Présentation du cadre de l'étude	25
1.5 PLANIFICATION COLLABORATIVE DE ROBOTS MOBILES	25
1.5.1 Formulation du problème pour un robot	25
1.5.2 Optimalité de la planification de trajectoire	25
1.5.3 Planification de trajectoire pour une flottille de robots	26
1.5.4 Approche discrète vs. approche continue	26
1.5.4.1 Les approches discrètes	26
1.5.4.2 Les approches continues	27
1.5.5 Présentation du cadre de l'étude	28
1.6 ORGANISATION DU MÉMOIRE DE THÈSE	28
1.6.1 Chapitre 2 - Boîte à outils	28
1.6.2 Chapitre 3 - Localisation d'un robot de type unicycle	29
1.6.3 Chapitre 4 - Localisation dans un cadre général	29
1.6.4 Chapitre 5 - Navigation autonome d'une flottille de robots	29
1.6.5 Chapitre 6 - Conclusion générale	29
1.7 LE FAUTEUIL ROULANT ÉLECTRIQUE AUTONOME	29
1.8 PUBLICATIONS	30
2 BOÎTE À OUTILS	31
2.1 MODÉLISATION DES ROBOTS MOBILES À ROUES	33
2.1.1 Définitions	33

2.1.2	Contraintes holonomes et non-holonomes	34
2.1.3	Commandabilité des systèmes non-holonomes	35
2.1.4	Classification des robots mobiles à roues	39
2.1.4.1	Hypothèse de travail	39
2.1.4.2	Notations utilisées	40
2.1.4.3	Les différents types de roues	40
2.1.4.4	Écriture des contraintes	41
2.1.4.5	Degré de mobilité et centre instantané de rotation	43
2.1.4.6	Classification des robots mobiles à roues	47
2.2	OBSERVABILITÉ, PLATITUDE ET ALGÈBRE DIFFÉRENTIELLE	51
2.2.1	Système et dynamiques	53
2.2.1.1	Système non linéaire	54
2.2.1.2	Entrées du système	54
2.2.1.3	Dynamique du système	55
2.2.1.4	Sorties du système	55
2.2.1.5	État et réalisation	55
2.2.2	Étude des propriétés d'un système non-linéaire	55
2.2.2.1	Platitude différentielle	56
2.2.2.2	Inversion	57
2.2.2.3	Liens entre inversion et platitude	57
2.2.2.4	Observabilité	57
2.2.3	Platitude dans le cas des robots mobiles à roues	59
2.2.3.1	Robot de type $(2, 0)$	59
2.2.3.2	Robot de type $(3, 0)$	60
2.2.3.3	Robot de type $(2, 1)$	60
2.2.3.4	Robot de type $(1, 1)$	61
2.2.4	Dérivation numérique	63
2.2.4.1	Introduction et état de l'art	63
2.2.4.2	Un exemple introductif	64
2.2.4.3	Annihilateur - Estimateur	65
2.2.4.4	Version continue de l'estimateur de dérivée	67
2.2.4.5	Version discrète de l'estimateur de dérivée	68
	CONCLUSION	69
3	LOCALISATION D'UN ROBOT DE TYPE UNICYCLE	71
3.1	MODÈLE DU ROBOT ET SON MODÈLE DE MESURE	73
3.1.1	Modèle cinématique du robot	73
3.1.2	Capteurs et modèle de mesure	73
3.2	LOCALISABILITÉ	75
3.2.1	Observabilité et Localisabilité	75
3.2.2	Localisabilité dans le cadre géométrique	75
3.2.3	Localisabilité dans le cadre algébrique	75
3.2.4	Exemple	76
3.3	QUELQUES CONDITIONS DE LA LOCALISABILITÉ	76
3.3.1	Présentation des cas d'étude	76
3.3.2	Un point d'intérêt (mesures : azimut et élévation)	77
3.3.2.1	Étude de la localisabilité	77
3.3.2.2	Estimateur d'état et des vitesses	78
3.3.3	Un point d'intérêt et une cible (mesures : azimuts et élévations)	79
3.3.4	Un point d'intérêt avec boussole (mesures : azimut, élévation et θ)	82

3.3.4.1	Étude de la localisabilité	82
3.3.4.2	Estimateur d'état et des vitesses	82
3.3.5	Deux points d'intérêt (mesures : deux élévations)	83
3.3.5.1	Étude de la localisabilité	83
3.3.5.2	Estimateur d'état et des vitesses	84
3.3.6	Deux points d'intérêt (mesures : deux azimuts)	86
3.3.6.1	Étude de la localisabilité	86
3.3.6.2	Estimateur d'état	88
3.3.7	Deux points d'intérêt (mesures : une élévation et un azimut)	89
3.3.7.1	Étude de la localisabilité	89
3.3.7.2	Estimateur d'état	90
3.3.8	Synthèse des résultats obtenus	91
3.4	SIMULATION ET EXPÉRIMENTATION	91
3.4.1	Paramètres de simulation et cas d'étude	92
3.4.2	Résultats numériques	92
3.4.3	Résultats expérimentaux	93
3.5	FUSION DANS LE CAS MULTI-POINTS D'INTÉRÊT	94
3.5.1	Hypothèse de travail	96
3.5.2	Étude de sensibilité de la solution proposée	96
3.5.3	Algorithme de fusion dans le cas de plusieurs points d'intérêt	97
3.5.4	Résultats en simulation	97
3.5.4.1	Simulation avec un bloc de points d'intérêt	97
3.5.4.2	Simulation avec deux blocs de points d'intérêt	99
	CONCLUSION	99
4	LOCALISATION DANS UN CADRE GÉNÉRAL	101
4.1	RAPPEL DES MODÈLES CINÉMATIQUES	103
4.1.1	Robot de type (2, 0)	103
4.1.2	Robot de type (3, 0)	103
4.1.3	Robot de type (2, 1)	103
4.1.4	Robot de type (2, 1)	103
4.2	LOCALISATION	104
4.2.1	Un point d'intérêt et une cible (mesures : azimuts et élévations)	104
4.2.1.1	Rappel du modèle de mesures	104
4.2.1.2	Localisation d'un robot de type (2, 0)	104
4.2.1.3	Localisation d'un robot de type (3, 0)	105
4.2.1.4	Localisation d'un robot de type (2, 1)	106
4.2.1.5	Localisation d'un robot de type (1, 1)	107
4.2.2	Un point d'intérêt avec boussole (mesures : azimut, élévation et θ)	108
4.2.2.1	Rappel du modèle de mesure	108
4.2.2.2	Localisation d'un robot de type (2, 0)	108
4.2.2.3	Localisation d'un robot de type (3, 0)	108
4.2.2.4	Localisation d'un robot de type (2, 1)	109
4.2.2.5	Localisation d'un robot de type (1, 1)	110
4.2.3	Deux points d'intérêt (mesures : deux élévations)	110
4.2.3.1	Rappel du modèle de mesure	110
4.2.3.2	Localisation d'un robot de type (2, 0)	111
4.2.3.3	Localisation d'un robot de type (3, 0)	111
4.2.3.4	Localisation d'un robot de type (2, 1)	112
4.2.3.5	Localisation d'un robot de type (1, 1)	113

4.2.4	Deux points d'intérêt (mesures : deux azimuts)	115
4.2.4.1	Rappel du modèle de mesure	115
4.2.4.2	Localisation d'un robot de type (2, 0)	115
4.2.4.3	Localisation d'un robot de type (3, 0)	115
4.2.4.4	Localisation d'un robot de type (2, 1)	116
4.2.4.5	Localisation d'un robot de type (1, 1)	117
4.2.5	Deux points d'intérêt (mesures : une élévation et un azimut)	118
4.2.5.1	Rappel du modèle de mesure	118
4.2.5.2	Localisation d'un robot de type (2, 0)	118
4.2.5.3	Localisation d'un robot de type (3, 0)	119
4.2.5.4	Localisation d'un robot de type (2, 1)	120
4.2.5.5	Localisation d'un robot de type (1, 1)	121
4.3	RÉSULTATS DE SIMULATION	121
4.3.1	Paramètres de simulation et cas d'étude	121
4.3.2	Résultats numériques	122
	CONCLUSION	122
5	NAVIGATION AUTONOME D'UNE FLOTTILLE DE ROBOTS	125
5.1	ALGORITHME DE PLANIFICATION DE TRAJECTOIRE	129
5.1.1	Rappel du modèle du robot	129
5.1.2	Planification de trajectoire : platitude et calcul variationnel	130
5.1.2.1	Chemins et trajectoires	131
5.1.2.2	Planification de trajectoire	131
5.1.2.3	Formulation mathématique	132
5.1.3	Résolution et implémentation en temps réel	133
5.2	LE BLOC STRATÉGIE	133
5.2.1	Choix de la fonction coût	134
5.2.1.1	Banque de fonctions coût	134
5.2.1.2	Détermination de la fonction coût par fusion	135
5.2.1.3	Influence des poids sur la planification	136
5.2.2	Choix de l'objectif	137
5.2.2.1	Calcul de la fonction de satisfaction	137
5.2.2.2	Algorithme de coopération	139
5.2.3	Modification des contraintes	140
5.2.4	Modification simultanée des contraintes et de la fonction coût	141
5.2.5	Sur l'influence des différents paramètres	142
5.3	RÉSULTATS D'EXPÉRIMENTATIONS ET DE SIMULATIONS	143
5.3.1	Définition des critères de validation	144
5.3.1.1	L'extensibilité	144
5.3.1.2	La complexité	144
5.3.1.3	Choix des paramètres	144
5.3.1.4	L'adaptabilité de l'algorithme	144
5.3.2	Résultats d'expérimentations et de simulations	145
5.3.2.1	Description de la plateforme	145
5.3.2.2	Résultats pour le critère d'extensibilité	146
5.3.2.3	Résultats pour le critère de complexité	146
5.3.2.4	Résultats concernant le choix des paramètres	148
5.3.2.5	Critère d'adaptabilité	148
	CONCLUSION	149

6	CONCLUSION GÉNÉRALE	151
6.1	CONCLUSION	151
6.2	PERSPECTIVES	152
A	ANNEXES	155
A.1	RAPPEL D'ALGÈBRE	157
A.1.1	Définitions générales	157
A.2	ÉTUDE DE L'OBSERVABILITÉ DANS LE CADRE GÉOMÉTRIQUE	160
A.3	CROCHET DE LIE	162
A.4	INTERSECTION DE DEUX CERCLES	163
A.5	ÉTUDE DE LA PLATITUDE À L'AIDE DE MAPLE	164
A.5.1	Platitude d'un robot de type (2, 0)	164
A.5.2	Platitude d'un robot de type (3, 0)	165
A.5.3	Platitude d'un robot de type (2, 1)	165
A.5.4	Platitude d'un robot de type (1, 1)	165
	BIBLIOGRAPHIE	167

LISTE DES FIGURES

1.1	Quelques exemples d'utilisation de robot	19
1.2	Architecture logicielle d'un robot mobile à roues	20
2.1	Notation utilisée et premier exemple de robot mobile	33
2.2	Notation pour l'étude d'une roue	36
2.3	Conventions de notations pour les roues	41
2.4	Différents types de roues utilisées pour les plates-formes mobiles.	41
2.5	Exemples et contre-exemple de centre instantané de rotation (CIR)	45
2.6	Exemple pour chacun des degrés de mobilité possibles	46
2.7	Exemple pour chaque degré de directionnalité	47
2.8	Robot de type (3, 0)	48
2.9	Le robot HILARE du LAAS-CNRS	49
2.10	Robot de type (2, 0)	49
2.11	Robot de type (2, 1)	50
2.12	Robot de type (1, 1)	50
2.13	Robot de type (1, 2)	51
3.1	Notations pour la localisation	74
3.2	Comparaison des algorithmes <u>NON-A-HM4</u> et <u>EKF-HM4</u>	93
3.3	Le robot - Une image de la caméra	95
3.4	Trajectoire utilisée pour les expérimentations	95
3.5	Localisation multi-points d'intérêt par fusion	98
3.6	Simulation avec un bloc de points d'intérêt	98
3.7	Simulation avec deux blocs de points d'intérêt	99

5.1	Architecture logicielle d'un robot mobile à roues	126
5.2	Horizons de calculs et de planification.	133
5.3	Modification de la fonction coût - effet sur la trajectoire	137
5.4	Modification de la fonction coût - effet sur la trajectoire	138
5.5	Surface pour le calcul de la fonction de congestion	139
5.6	Coopération avec changement d'objectif - effet sur la trajectoire . .	141
5.7	Modifications des limitations de vitesse - effet sur la trajectoire . .	142
5.8	Changement simultané de la fonction coût et des contraintes - Effet sur la trajectoire	143
5.9	Carte des environnements utilisés pour les simulations et pour les expérimentations	147
5.10	Temps de navigation en fonction du nombre de robots dans la flottille, $\beta = 5, \alpha = 1$	147
5.11	Temps d'optimisation moyen en fonction du nombre de robots . .	148

NOTATIONS

$\mathcal{R}_i = (0_i, \vec{x}_i, \vec{y}_i, \vec{h}_i)$	repère de l'espace
$\mathbf{p} = (x, y, \theta)^T$	posture du robot dans un repère \mathcal{R} fixe lié à l'environnement
z_A	affiche du point tel que $z_A = x_A + iy_A$
$\mathcal{R}(z_A) = x_A$	partie réelle du complexe z_A
$\mathcal{I}(z_A) = y_A$	partie imaginaire du complexe z_A
$\overline{z_A}$	complexe conjugué de z_A
z	affiche du robot dans le plan de mouvement
A_ℓ	point d'intérêt A_ℓ d'affiche z_{A_ℓ} (dans le plan)
$z_{r,\ell} = z_l - z$	affiche relatif entre le robot et un point d'intérêt A_ℓ
$(\alpha_\ell, \beta_\ell)$	angles relatifs robot - point d'intérêt A_ℓ
$\mathbf{q} = (\mathbf{q}_1, \dots, \mathbf{q}_i, \dots, \mathbf{q}_n)^T$	vecteur de configuration
\mathcal{N}	espace de configuration
v	vitesse linéaire
ω	vitesse angulaire
$\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_i, \dots, \mathbf{g}_n)^T$	champ de vecteur
$d.l$	degré de liberté
\mathcal{A}	Algèbre de Lie d'accessibilité
r	rayon de la roue
P	centre de la roue
Q	point de contact entre la roue et le sol
φ	angle de rotation de la roue
γ	angle de rotation de la roue par rapport au robot
d	décentrement de la roue par rapport au cadre du robot
β	rotation de la roue par rapport à l'axe vertical
L	distance entre deux roues
N	nombre de roues
$A = (x_A, y_A, h_A)$	point A et ses coordonnées dans un repère \mathcal{R}
δ_s	degré de manœuvrabilité
δ_m	degré de mobilité
L, K, F	anneau ou corps
$P(X, \dots, X^n)$	polynôme
$X^{(n)}$	dérivée n^{ieme} de X par rapport au temps
$P(X, \dots, X^{(n)})$	polynôme différentiel
$d^o \text{ tr}$	degré de transcendance algébrique
$d^o \text{ tr diff}$	degré de transcendance différentielle
$\Theta = \exp(i\theta)$	nombre complexe associé à θ
$B = \exp(i\beta)$	nombre complexe associé à B
$\Gamma = \exp(i\gamma)$	nombre complexe associé à γ

\mathbf{u}	commande d'un système
\mathbf{x}	état d'un système tel que $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
$\mathbf{y} = (y_{m1}, \dots, y_{mn})$	mesure telle que $\mathbf{y} = \mathbf{h}(\mathbf{x})$
\mathbf{y}_{aux}	sortie auxiliaire construite à partir des sorties mesurées
\mathbf{z}	sortie du système
s	variable de Laplace
$\Pi_{k,\mu}^{N,n}$	annihilateur de paramètres k, μ, N, n
\mathcal{C}_ℓ	cercle de centre \mathcal{O}_ℓ et de rayon r_ℓ
y_{mi}	variable mesurée
y_{fi}	variable filtrée
\hat{y}	variable estimée

INTRODUCTION



SOMMAIRE

1.1	ROBOTS ET ROBOTIQUE DE SERVICE	17
1.1.1	Un robot, trois visions françaises différentes	17
1.1.2	Définition d'un robot	17
1.1.3	Robotique de service : un secteur en pleine croissance	18
1.1.4	Périmètres d'application	18
1.2	LA ROBOTIQUE MOBILE AUTONOME	18
1.2.1	Un premier exemple de robot autonome	18
1.2.2	Architecture d'un robot mobile autonome	20
1.3	LES CHALLENGES SCIENTIFIQUES DE CE TRAVAIL	21
1.4	LOCALISATION DE ROBOTS MOBILES	22
1.4.1	Vidéo : détection des points d'intérêt	23
1.4.2	Correspondance entre observations réelles et attendues	23
1.4.3	Bref état de l'art sur la localisation	24
1.4.4	Présentation du cadre de l'étude	25
1.5	PLANIFICATION COLLABORATIVE DE ROBOTS MOBILES	25
1.5.1	Formulation du problème pour un robot	25
1.5.2	Optimalité de la planification de trajectoire	25
1.5.3	Planification de trajectoire pour une flottille de robots	26
1.5.4	Approche discrète vs. approche continue	26
1.5.5	Présentation du cadre de l'étude	28
1.6	ORGANISATION DU MÉMOIRE DE THÈSE	28
1.6.1	Chapitre 2 - Boîte à outils	28
1.6.2	Chapitre 3 - Localisation d'un robot de type unicycle	29
1.6.3	Chapitre 4 - Localisation dans un cadre général	29
1.6.4	Chapitre 5 - Navigation autonome d'une flottille de robots	29
1.6.5	Chapitre 6 - Conclusion générale	29
1.7	LE FAUTEUIL ROULANT ÉLECTRIQUE AUTONOME	29
1.8	PUBLICATIONS	30

LA robotique mobile est un champ d'étude très vaste tant scientifique que sociétal et économique. Cette introduction va tenter de présenter quelques exemples de robots répondant à des attentes différentes en fonction du public ciblé. Nous donnerons ensuite quelques chiffres qui montrent bien les potentialités

de la robotique de service tant auprès des industriels que des particuliers. Cependant ces potentialités se confrontent à des problématiques scientifiques nombreuses et pluridisciplinaires. Nous présenterons donc le sous-domaine de la robotique auquel nous souhaitons apporter une contribution, la robotique mobile autonome. La présentation des concepts généraux nous permettra d'extraire deux problématiques principales, la localisation et la planification de trajectoire, qui seront explicitées au regard de la littérature du domaine. Nous soulignerons alors les contributions principales de ce mémoire du point de vue scientifique.

1.1 ROBOTS ET ROBOTIQUE DE SERVICE

1.1.1 Un robot, trois visions françaises différentes

Selon qu'on demande à des chercheurs, des industriels, ou au grand public ce qu'est un robot, les réponses obtenues seront foncièrement différentes. Pour le grand public français, la robotique se limite souvent à des images de films comme *I-Robot* ou *Terminator* et sont souvent associés à une ambiance de fin du monde. Cette vision apocalyptique a quelque peu changé depuis la mise sur le marché du robot aspirateur *Roomba* de la société *I-Robot* ou encore des tondeuses automatiques, popularisant l'image d'un robot plus agréable et plus utile. Pour l'industriel, la robotique signifie souvent robotique industrielle destinée à l'automatisation de chaînes de production. La robotique dite de service commence depuis une dizaine d'années son essor mais n'est pour l'instant qu'à un stade préliminaire. Pour le chercheur, les choses sont plus complexes. Il suffit de se rendre aux journées nationales de la recherche en robotique (en France), organisées par le GDR Robotique, pour se rendre compte du large spectre des problèmes de recherche et des domaines d'application de la robotique. Notons la robotique médicale, humanoïde, la nano-robotique, la bio-robotique et finalement la robotique mobile.

1.1.2 Définition d'un robot

Quelles que soient les applications finales désirées, peut être considérée comme robot toute machine intelligente qui peut effectuer des tâches nécessitant en temps normal une intervention humaine. C'est une définition très large mais qui couvre bien les bases de la robotique. Un robot n'est pas juste un périphérique distant commandé par l'humain (comme l'est une voiture radio-commandée), mais il doit comporter des éléments d'intelligence artificielle. La robotique est donc un domaine où l'on cherche à créer et utiliser des entités réelles ayant les capacités suivantes :

- Une capacité d'action, à travers ses actionneurs,
- Une capacité de perception à travers ses capteurs,
- Une capacité d'interaction avec l'environnement,
- Une capacité de décision.

et ce dans l'optique d'accomplir une tâche avec ou sans l'intervention d'un humain.

Il faut donc pour obtenir un robot intégrer les quatre composantes suivantes :

- La conception, la modélisation et l'analyse de la problématique et des nouveaux besoins (performance, communication, adaptabilité à l'environnement, ...),
- La chaîne de perception, composée des capteurs qui « nourrissent » les robots en informations de natures diverses. Il faut conditionner et traiter ces informations pour en extraire une information pertinente et utile au robot,
- La chaîne d'action et de décision qui comporte plusieurs chaînons (cognitif, la planification de tâches, la planification de mouvements, la commande et les actionneurs),
- L'interaction avec l'environnement à travers la collaboration robots/robots, hommes/robots ou monde physique/robots.

1.1.3 Robotique de service : un secteur en pleine croissance

La robotique de service (à différencier de la robotique industrielle) ne relève pas que du domaine de la recherche ; en effet, d'après la Japanese Robotics Association, le marché de la robotique dédiée aux services à la personne devrait atteindre mondialement un chiffre d'affaire de 12 milliards de dollars en 2010 et environ 35 milliards en 2025 (chiffre 2009). D'après ABI Research, le marché de la robotique personnelle devrait atteindre 19 milliards de dollars en 2017 (chiffre 2010). Finalement d'après l'IFR - World Robotics 2009, la robotique de service devrait passer d'un marché de 13,5 milliards de dollars en 2009 à 24,3 milliards de dollars sur la période 2009 – 2012 et 100 milliards de dollars en 2020. En Europe, le marché de la robotique de service est estimé à 5 milliards de dollars (source IFR), soit 33% du marché. L'Allemagne est leader avec 51% du marché, suivie de l'Italie à 22% et de la France à 12%, soit 600 millions de dollars (source Syrobo, le syndicat français de la robotique de service). Tout ces chiffres amènent à la même conclusion : ce marché est en plein essor et sera un moteur de l'économie dans les années à venir.

1.1.4 Périmètres d'application

Le périmètre précis de la robotique de service est difficile à définir. Toujours d'après Syrobo, la robotique de services est le secteur à plus fort potentiel avec une forte croissance annuelle à deux, voire trois chiffres dans les prochaines années. Les grands champs d'application de la robotique de service sont entre autres (voir la figure 1.1 pour une illustration de ces exemples) :

- L'internet des objets et les maisons intelligentes (robot de télésurveillance),
- La santé (rééducation et assistance à la personne),
- L'éducation et les technologies (robot d'éducation du Korea's Institute),
- Les loisirs numériques (guide pour les musées),
- La surveillance et la sécurité,
- La coopération Homme-Robot dans les entreprises de services.

Les défis techniques et technologiques touchent un champ très large des sciences, notamment l'électronique, l'automatique et l'informatique, la mécanique, le design, les télécommunications. Dans ce travail, nous étudierons certaines problématiques liées à l'automatique et à l'informatique appliquées à la robotique mobile (robot à roues principalement).

1.2 LA ROBOTIQUE MOBILE AUTONOME

1.2.1 Un premier exemple de robot autonome

Comme cela vient d'être indiqué, nous concentrons notre étude sur l'apport de l'automatique et de l'informatique au service des robots mobiles à roues. Comme l'indique le nom, un robot à roues est une architecture mécanique rigide sur laquelle est fixé un certain nombre de roues (certaines motorisées, d'autres non) permettant le déplacement de cette architecture. L'exemple le plus courant est une voiture pour laquelle la commande de la vitesse et de la direction des roues motrices serait confiée à un ordinateur. L'objectif de ce travail est donc de rendre autonome un robot mobile à roues. C'est-à-dire que, pour un robot donné (une voiture par exemple), nous souhaitons développer des algorithmes permettant de



FIGURE 1.1 – *Quelques exemples d'utilisation de robot*

rendre ce robot autonome dans ses déplacements. L'humain choisit la mission à accomplir, le robot effectuant alors le reste du travail.

Reprenons le robot de surveillance donné en illustration de la robotique de service. Un tel robot doit être capable de circuler dans la zone à surveiller sans l'aide d'un humain. Pour mener à bien sa mission, il cartographie son environnement, il se localise dans sa carte, et il navigue entre des points de passage donnés par le superviseur. S'il détecte une intrusion, il déclenche une alarme pour avertir l'humain responsable.

Cet exemple met bien en évidence les principales problématiques de la robotique mobile :

- La cartographie et la localisation,
- La compréhension et l'adaptation dans un langage robot d'une mission assignée par un humain,
- La planification et le suivi de trajectoire,
- L'application des trajectoires planifiées,

- La perception d'éléments de l'environnement utiles à la mission.

Afin que le robot puisse répondre à ces problématiques, il a besoin d'une architecture logicielle dédiée qui est présentée dans la suite.

1.2.2 Architecture d'un robot mobile autonome

Pour pouvoir se déplacer de façon autonome sans entrer en collision et mener à bien la mission qui lui a été confiée, le robot mobile doit être doté d'une architecture logicielle adéquate. Une proposition d'architecture (relativement classique mais permettant de bien visualiser les différentes problématiques de la robotique mobile autonome) est donnée dans la figure 1.2. Elle met en évidence les principaux défis à relever pour mener à bien un tel projet.

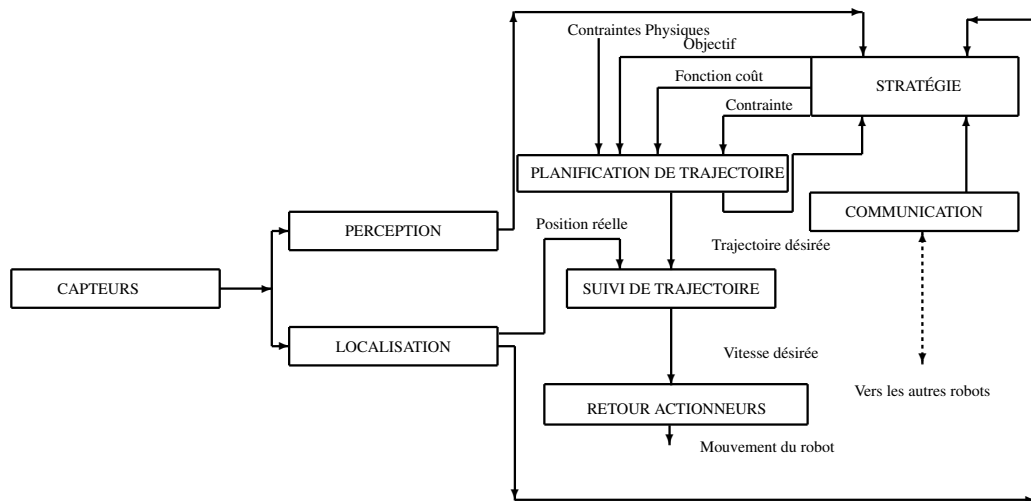


FIGURE 1.2 – Architecture logicielle d'un robot mobile à roues

Reprenons en détail les différents blocs pour bien les expliciter

Les capteurs sont les organes de perception du robot. Il existe deux classes de capteurs, les capteurs proprioceptifs donnant des renseignements sur l'état du robot (par exemple les capteurs de position ou de vitesse des roues, les capteurs de charge de la batterie) et les capteurs extéroceptifs qui renseignent sur l'état de l'environnement, c'est-à-dire ce qui est extérieur au robot lui-même. Les capteurs proprioceptifs sont utiles pour l'autonomie même du robot alors que les capteurs extéroceptifs permettent de donner au robot des informations utiles à la réalisation de sa mission. Ainsi, par exemple, les robots destinés à intervenir dans un environnement hostile à l'homme (centrale nucléaire, usine chimique, ...), seront équipés de capteurs permettant de **percevoir** la situation courante. Ce peut être une caméra, un capteur de température, de pression, un compteur Geiger, ... Ces capteurs, aussi importants soient-ils pour la mission, ne permettent pas de rendre le robot autonome. En effet, le robot n'est autonome qu'à condition de posséder une carte de l'environnement et d'être capable de se localiser dans cette carte. Les capteurs permettant de résoudre cette problématique ne sont pas légion. Citons tout de même le bien connu GPS pour les environnements extérieurs. Pour les environnements intérieurs il faudra utiliser un dispositif permettant de construire des relations géométriques entre le robot et son environnement. Citons les capteurs laser, les caméras, les capteurs radio ...

La localisation du robot susmentionné consiste à indiquer où le robot se situe dans une carte préétablie de l'environnement (remarquons qu'il est possible de construire la carte en même temps que le robot se localise, cette problématique étant appelée SLAM pour *Simultaneous Localisation And Mapping*, cartographie et localisation simultanées).

La perception du robot peut être décomposée en deux éléments. Le premier consiste à construire une carte de l'environnement grâce aux capteurs embarqués, le deuxième à percevoir les éléments de l'environnement nécessaires au bon accomplissement de la mission (par exemple détecter une intrusion pour un robot de surveillance).

La stratégie permet au robot de choisir une mission (globale ou intermédiaire). Cette mission peut être déterminée en concertation avec les autres robots (pour une flottille), avec un PC de contrôle distant ou de façon autonome par le robot (par exemple se diriger vers les points de plus fortes pressions/radiations ...).

La planification de trajectoire permet au robot, une fois la mission choisie, de planifier une trajectoire lui permettant de se rendre vers l'objectif désiré. La communication entre les deux derniers blocs (planification de trajectoire et stratégie) permet de choisir à la fois l'objectif et le comportement désirés pour se rendre à sa position finale désirée. De par notre expertise et en fonction des solutions que nous proposons pour la planification de trajectoire, nous avons décidé d'utiliser une fonction coût permettant de décider si le robot va essayer d'atteindre sa cible le plus rapidement possible, de minimiser l'énergie consommée, de limiter les risques de collisions ... De même un certain nombre de contraintes variables dans le temps peut être pris en compte comme par exemple des limitations de vitesse ... Le bloc idéal de planification de trajectoire doit aussi prendre en compte les contraintes physiques du robot. Il ne faut pas par exemple planifier une trajectoire où le robot doit rouler à 50km/h alors que physiquement il ne peut pas aller à plus de 30km/h .

Le bloc de suivi de trajectoire et le retour actionneur permettent, une fois la trajectoire désirée calculée, de l'appliquer sur le robot. A chaque instant, le robot vérifie s'il est bien arrivé au bon point de sa trajectoire calculée et éventuellement corrige la commande sur les roues pour compenser les erreurs.

1.3 LES CHALLENGES SCIENTIFIQUES DE CE TRAVAIL

Dans ce travail nous allons nous intéresser principalement au bloc **localisation** et aux blocs **stratégie et planification de trajectoire**. Ces trois blocs sont primordiaux et discriminants dans de nombreuses applications de robotique mobile autonome. Le champ des méthodes applicables à ces deux problématiques est très vaste, et une méthode donnant de bons résultats dans une situation pourrait donner de mauvais résultats dans une autre. On retrouve dans ces trois problématiques une partie des discussions qui alimentent les conférences dans le domaine et qui en font un domaine de grand foisonnement intellectuel. Par exemple vaut-il mieux une approche déterministe ou probabiliste pour la localisation de robot mobile ? Ou encore : faut-il représenter l'environnement du robot sous une forme

discrète ou continue pour la planification de trajectoire ? Ce mémoire, reflétant le travail de ses auteurs sera forcément enclin à donner une réponse à ces différentes questions. Afin de ménager « le suspens » nous allons pour le moment uniquement donner les critères qui nous semblent indispensables dans cette étude :

- La robotique est un domaine applicatif, les algorithmes et solutions proposés doivent pouvoir s’appliquer sur des robots réels. Cela implique de travailler avec des capteurs existants, même si perfectibles. Qui plus est, la complexité combinatoire sera un élément de choix entre différentes méthodes.
- La contrainte du robot réel impose la présence de perturbations et de contraintes physiques. Un algorithme demandant à une voiture de faire un mouvement de translation n’est pas très proche de la réalité physique d’un tel véhicule.
- Il existe une multitude de robots mobiles à roues. L’objectif est de proposer les algorithmes les plus génériques possibles, pouvant ainsi s’étendre à une large classe de problèmes.

Dans la suite de ce chapitre nous présenterons un état de l’art sur les deux problématiques que nous souhaitons étudier. La localisation de robots mobiles est un champ déjà largement étudié où les méthodes sont très nombreuses. Deux principales classes de méthodes ressortent, les méthodes relatives (type odométrie) et les méthodes globales (type GPS). Pour chaque type de méthode, deux principales écoles permettent d’étudier la problématique, l’école probabiliste et l’école déterministe. Un élément intéressant, encore peu étudié dans la littérature, est la capacité de la méthode à prédire, pour un type de robot et pour un type de capteur donné, s’il y a une solution à la problématique de localisation. Nous présenterons quelques éléments d’étude de la littérature et présenterons les points où nous souhaitons, par ce travail, apporter un nouvel éclairage. Concernant la planification de trajectoire, un élément discriminant déjà présenté consistait à séparer les méthodes représentant l’environnement comme un ensemble discret et celles le représentant comme un ensemble continu. Après avoir rappelé la problématique générale nous présenterons quelques travaux appartenant à chacune de ces catégories.

1.4 LOCALISATION DE ROBOTS MOBILES

La question principale est « Where am I ? », (« Où suis-je ? ») du titre d’une publication de J. Borenstein en 1996 [14]. Cette publication couvre le vaste champ de cette problématique en allant du capteur aux algorithmes de localisation et de cartographie. De façon un peu plus détaillée, l’objectif d’un algorithme de localisation est de pouvoir estimer la position du robot (notée $(x, y, \theta)^T$ en général) dans un référentiel donné (soit le référentiel global soit un référentiel lié au point de départ du robot). Cette estimation doit être aussi précise que possible en fonction des mesures disponibles.

Si l’on se restreint à la navigation en intérieur (pour la navigation extérieure, les techniques utilisées sont quelque peu différentes car il est plus facile de coupler un capteur GPS avec d’autres capteurs pour obtenir une bonne précision), trois principales classes de techniques ressortent : la navigation sans carte, la navigation avec construction de carte simultanée et la navigation basée sur une carte ou sur des points repérables de coordonnées connues. C’est cette dernière classe que nous souhaitons étudier dans ce travail. Afin de localiser le robot, il est nécessaire d’acquérir des informations capteurs afin de détecter des points d’intérêt dans les informations obtenues et d’établir une correspondance entre les observations

réelles et les observations attendues. Celle-ci sera utilisée pour calculer la position du robot [34].

1.4.1 Vidéo : détection des points d'intérêt

Pour une caméra il faut être capable de récupérer le flux vidéo, ce qui est fait *via* un protocole informatique dédié. Une fois le flux accessible, il est nécessaire de chercher les points d'intérêt dans l'image. Un point d'intérêt est un point de l'espace que le robot est capable de distinguer et de différencier d'autres points avec ses capteurs embarqués. Cette distinction doit être répétable et doit pouvoir s'effectuer au cours du temps. Différentes méthodes sont utilisables, notons cependant les plus courantes, SIFT [90], SURF [7], les points d'angles ([63, 139]).

1.4.2 Correspondance entre observations réelles et attendues

Il s'agit de comparer les observations réelles avec celles que le robot aurait obtenues s'il était réellement là où il pensait être. C'est typiquement le schéma de travail du filtre de Kalman Étendu que nous allons présenter ici.

S. Chen [22] donne une vue d'ensemble de l'utilisation du filtre de Kalman pour la robotique depuis plus d'une trentaine d'années. Ainsi [29, 25] utilisent le filtre de Kalman pour la localisation de robots mobiles. Rappelons ici les équations principales du filtre de Kalman.

Le filtre de Kalman Étendu est une évolution du filtre de Kalman pour l'étude des systèmes non linéaires. En fait, le filtre de Kalman Étendu peut être vu comme une version locale du filtre de Kalman, ce qui signifie que le modèle d'état et le modèle de mesure sont linéarisés localement autour de l'estimation courante (voir par exemple [20] et [147]). Il n'est donc pas nécessaire que le modèle d'état et le modèle de mesure soient linéaires, cependant ils doivent être des fonctions différentiables par rapport aux variables d'état. Prenons $\mathbf{p}_k = (x_k, y_k, \theta_k)^T$ le vecteur d'état à l'instant $t_k = kT_s, k \in \mathbb{N}$ (T_s étant la période d'échantillonnage), \mathbf{u}_k est le vecteur de commande à l'instant t_k . Le modèle cinématique du robot permet de donner \mathbf{p}_{k+1} en fonction de \mathbf{p}_k et de la commande \mathbf{u}_k de la façon suivante $\mathbf{p}_{k+1} = \mathbf{f}(\mathbf{p}_k, \mathbf{u}_k)$. Le modèle de mesure en temps discret étant :

$$\mathbf{y}_m = \mathbf{h}(\mathbf{p}_k) = \mathbf{y}_k + \varpi_k, \quad (1.1)$$

où \mathbf{y}_k est la sortie du système, mesurée à l'instant t_k , ϖ_k est le bruit de mesure (gaussien de moyenne nulle et de matrice de covariance R_k). La fonction \mathbf{f} est la fonction d'état du système, elle est utilisée pour prédire l'état à l'instant t_{k+1} à partir de l'état à l'instant t_k . La fonction \mathbf{h} est utilisée pour prédire l'observation que le robot devrait faire s'il était réellement là où il le « pense » à l'instant t_{k+1} . La différence entre cette prédiction et la mesure réelle, appelée innovation, est utilisée pour donner une estimation finale de la position du robot. Les fonctions \mathbf{f} et \mathbf{h} étant non-linéaires, leur matrice jacobienne est calculée à chaque pas de calcul pour la linéarisation des équations.

Sans même rentrer dans les détails de l'algorithme, plusieurs inconvénients de cette méthode peuvent être soulignés.

1. Il est nécessaire de connaître une initialisation de l'état, en effet cette initialisation est nécessaire pour l'étape de prédiction,
2. Il est nécessaire de connaître les propriétés statistiques du bruit. Qui plus est, la nature même du bruit est fixée par la méthode. Or par définition de

la notion de bruit, il est toujours difficile d'avoir ce genre d'informations qui peuvent en plus varier au cours du temps en fonction par exemple des conditions de luminosité pour une caméra,

3. La mise à jour de l'estimation se base sur une linéarisation, à l'ordre 1, des différentes fonctions non-linéaires du système. Or, dans certains cas, cette linéarisation est non observable alors même que le non linéarisé est observable.

1.4.3 Bref état de l'art sur la localisation

La dernière étape nécessaire à un système de localisation est celle qui va nous intéresser dans ce travail. Elle tente de répondre à la question suivante :

Comment, à partir d'observations relatives à des points connus de la carte, le robot peut-il se localiser ?

Notons que nous nous intéressons ici aux méthodes de localisation dites absolues, en opposition avec les méthodes dites relatives où le robot se localise grâce aux capteurs proprioceptifs, tels l'odométrie ou la centrale inertielle. Les méthodes relatives souffrent des problèmes liés notamment au glissement des roues sur le sol qui les rendent inutilisables du moins comme unique solution de localisation. Ces approches doivent donc être complétées par des méthodes dites absolues qui vont pouvoir à intervalle régulier donner une position du robot dans son environnement (rappelons que si l'intervalle est trop grand pour l'application désirée les méthodes relatives peuvent être utilisées entre deux mises à jour de la méthode absolue). L'approche la plus connue pour la localisation absolue est le GPS (qui ne fonctionne qu'en extérieur). Le principe de fonctionnement du GPS (de manière très simplifiée) consiste à mesurer la distance relative entre le capteur et au moins quatre satellites. Connaissant la position des satellites, il est possible de localiser le capteur et donc le robot (cette localisation se fait par intersection des cercles centrés sur les satellites). Bien qu'inutilisable en environnement intérieur, il est intéressant de remarquer que le problème de localisation se résume à un problème de géométrie. En environnement intérieur, les satellites sont remplacés par des points d'intérêt. Un point d'intérêt est donc un point de l'espace de coordonnées connues ou inconnues. Qui plus est il peut être actif ou naturel. Un ensemble de points d'intérêt actifs de coordonnées connues est formé par exemple par les bornes wifi d'un bâtiment [12]. De même [123] utilise le protocole *Bluetooth* alors que [156] utilise des *patterns* magnétiques. L'un des aspects les plus réhébilitaires de ces méthodes est la nécessité de modifier l'environnement pour les faire fonctionner. *A contrario* les points d'intérêt naturels de coordonnées connues sont des points distinguables par le robot *via* par exemple sa caméra (points SIFT, SURF, ...) ou encore des organisations géométriques particulières de l'espace récupérables par un capteur laser. L'idée principale étant d'utiliser la géométrie, et à condition qu'il y ait suffisamment de points disponibles, les méthodes à bases de triangulation sont bien adaptées [52]. Cependant dans certaines situations il est difficile de trouver assez de points d'intérêt. Avec un seul point d'intérêt il est alors nécessaire de calculer un facteur d'échelle, qui peut être issu de la forme du point d'intérêt [69, 80, 130] ou alors du mouvement du robot [4, 31] où l'odométrie est utilisée pour connaître le mouvement du robot entre deux instants successifs (avec les problèmes de glissement éventuels).

L'un des aspects non abordés dans les différentes méthodes présentées est celui d'un critère objectif permettant, avant même l'expérimentation, de savoir si le ro-

bot est oui ou non localisable pour un ensemble de capteurs donné. L'un des cadres de travail idéaux pour cette étude est lié à la théorie de l'observabilité au sens de la communauté automatique. Cette étude a en partie été menée dans les travaux [11, 28, 99, 8, 13] en utilisant l'approche géométrique de l'observabilité des systèmes non linéaires. Un des inconvénients de cette approche géométrique est que le critère permettant d'aboutir à la non observabilité est très coûteux en terme de temps de calcul. Qui plus est les travaux précédents se sont concentrés uniquement sur l'étude du robot unicycle utilisant des mesures angulaires seules (typiquement celles données par une caméra par rapport à un point d'intérêt).

1.4.4 Présentation du cadre de l'étude

Nous allons dans ce travail présenter un cadre d'étude algébrique qui semble plus pertinent (que le cadre géométrique) pour l'étude de l'observabilité des systèmes non-linéaires. En effet, la capacité de l'approche géométrique à donner un critère de non-localisabilité n'est pas clairement établie et dans tous les cas, le coût combinatoire d'un tel critère est rédhibitoire. Qui plus est, l'écriture algébrique des équations permet de simplifier l'expression d'un tel critère. Finalement, l'approche proposée permet de passer très facilement du critère de localisabilité à la construction de l'estimateur d'état. Ce cadre d'étude sera utilisé pour effectuer une étude de la localisabilité de quatre des cinq classes de robots mobiles à roues (en effet [19] prouve que tout robot mobile à roues appartient forcément à une des cinq classes possibles, chaque classe ayant un modèle cinématique associé) pour l'ensemble des capteurs donnant une information géométrique (angle ou distance) par rapport à des points d'intérêt.

1.5 PLANIFICATION COLLABORATIVE DE ROBOTS MOBILES

1.5.1 Formulation du problème pour un robot

La planification pour un robot se formule de la façon suivante [40]. Considérons un environnement donné (étude effectuée dans le plan). Le robot est dans la configuration $I = (x_I, y_I, \theta_I)$ et nous voulons qu'il aille dans la configuration $F = (x_F, y_F, \theta_F)$. L'objectif est de calculer une trajectoire sans collision entre I et F . Formellement, nous cherchons une courbe continue du plan avec un des points extrêmes de la courbe sur le point de départ I et l'autre extrémité sur le point final F tenant compte des capacités de déplacement du robot et du but de sa mission.

1.5.2 Optimalité de la planification de trajectoire

Nous pouvons remarquer que ce problème est dans le cas général extrêmement compliqué. En effet, il est nécessaire de travailler dans un espace continu pour trouver une courbe continue de cette espace. Qui plus est, étant donné la continuité de l'espace et de la courbe, quand le problème est solvable, il existe une infinité de trajectoires. Ainsi l'algorithme de planification de trajectoire idéal serait capable de dire s'il existe ou non une solution au problème et s'il en existe une, il devrait être capable de donner la trajectoire optimale. Notons que la notion même d'optimalité prête à discussion. Par exemple, quelle trajectoire est-elle optimale ? Est-ce la trajectoire la plus courte en distance ou la plus courte en temps de navigation, faut-il chercher l'optimalité en termes de temps de calcul ?

1.5.3 Planification de trajectoire pour une flottille de robots

Au delà de la difficulté intrinsèque du problème, différentes nouvelles problématiques viennent s'ajouter lorsque l'on souhaite planifier la trajectoire d'une flottille de robots dans un environnement éventuellement dynamique. Une flottille de robots est un ensemble de robots ayant chacun une tâche de navigation propre. Pour mieux profiter des interactions dans un système multi-robots il est nécessaire d'y introduire des mécanismes de coordination afin de rendre cohérentes entre elles les actions des robots. En étudiant les communautés humaines, [103] a identifié trois processus de coordination :

- la coordination par ajustements mutuels où les individus s'accordent pour partager des ressources en vue d'atteindre un but commun (aucun individu n'a de contrôle sur les autres individus, et le processus de décision est conjoint),
- la coordination par "leadership" ou supervision où une relation hiérarchique existe entre des individus (certains individus exercent alors un contrôle sur d'autres),
- la standardisation où des procédures sont prédéfinies en vue de situations d'interactions particulières (par exemple des règles dont l'application peut limiter les conflits).

Les différents mécanismes de coordination exploités dans les systèmes multi-robots sont tous des manifestations d'un ou de plusieurs de ces processus fondamentaux. Les deux principales approches appliquées à la planification de trajectoire sont l'approche centralisée, dans laquelle les trajectoires de la flottille sont générées via un superviseur, et l'approche décentralisée où chaque robot calcule sa propre trajectoire en échangeant des informations avec ses voisins. L'avantage de la première solution est qu'il est plus facile de générer des comportements complexes (par exemple pour l'exploration d'environnement, pour le transport de charge, ...) qu'avec une flottille sans meneur. *A contrario*, le gros défaut de ces méthodes est que si pour une raison ou une autre, le robot meneur est rendu indisponible durant la mission, ou que la communication entre les suiveurs et le meneur est impossible, c'est l'ensemble des missions des robots qui est compromis. Les approches sans meneur sont donc une solution à cette problématique mais avec une difficulté supplémentaire pour l'attribution des missions et l'accomplissement de tâches de navigation complexes.

1.5.4 Approche discrète vs. approche continue

Avec ou sans meneur, les deux approches modifient plus la façon dont seront choisis les objectifs que la façon de planifier la trajectoire. Il existe encore une fois deux écoles de travail, celle où l'espace est discrétisé et celle où l'espace est continu.

1.5.4.1 Les approches discrètes

La première école s'appuie sur les méthodes développées en intelligence artificielle. Elle consiste à construire un graphe de toutes les positions atteignables par le robot (le déplacement entre deux points étant un déplacement dit unitaire, aller en avant, aller en arrière, à gauche, à droite) dans l'environnement considéré. Ensuite, un algorithme d'optimisation sur un graphe est utilisé, comme, pour ne citer que les plus connus, A* [64], Dijkstra [35], l'algorithme de parcours en profondeur

[41], l'algorithme de parcours en largeur [74], l'algorithme de recherche best-first [117]. L'un des désavantages de ces approches est la nécessité de connaître tout l'environnement au début de l'optimisation du graphe. De plus, en général, ces algorithmes ne prennent pas en compte les contraintes physiques des robots. Dans [142, 141, 140] l'algorithme de recherche A* est étendu au cas où l'environnement est dynamique. Plus récemment, l'algorithme RRT, (voir [78, 1, 149]) a été utilisé avec succès dans les environnements dynamiques.

1.5.4.2 Les approches continues

Les approches continues permettent de résoudre un certain nombre de problématiques des approches discrètes, notamment la difficulté de choisir le pas d'échantillonnage du graphe. En effet, un pas d'échantillonnage trop grand risque d'entraîner des mouvements unitaires impossibles à effectuer réellement par le robot comme dans le cas où un obstacle, plus petit que le pas, se trouve entre le robot et le point suivant à atteindre. *A contrario*, un pas d'échantillonnage trop petit fait exploser le coût combinatoire de l'optimisation du graphe, rendant la méthode réductible pour une utilisation temps réel. L'idée principale des approches continues est de modéliser l'espace des configurations comme un ensemble continu et de chercher une trajectoire dans cet espace. Les méthodes les plus marquantes dans cette approche sont :

- Les approches par puits de potentiels [3, 17],
- Les approches par histogramme de champ de vecteur [15, 16, 148] dont les concepts sont comparables à ceux donnés dans la méthode précédente,
- Les algorithmes Bug [93, 92, 70, 94].

Nous présentons un peu plus en détail la première approche, pour plus de détails sur les autres approches se référer aux références citées.

Les puits de potentiels [3, 17] Dans cette approche, le robot est considéré comme une particule qui agit sous l'influence d'un champ de potentiel U modulé en fonction de la structure de l'espace libre. Les obstacles sont considérés comme des champs de potentiel répulsif alors que l'objectif est considéré comme un champ de potentiel attractif. Ainsi le potentiel global est le suivant :

$$U(q) = U_{objectif}(q) + \sum U_{obstacles}(q).$$

La force résultante est alors donnée par :

$$F = -\nabla U(q) = - \begin{pmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{pmatrix}.$$

Il ne reste plus qu'à appliquer sur le robot un petit mouvement qui suit localement la force calculée. Pour implémenter des comportements différents pour le robot il faut trouver les bonnes fonctions de potentiel qui permettront d'obtenir le comportement désiré (l'approche la plus simple consiste à prendre des potentiels paraboliques).

Une critique et mise en avant des points faibles de cette méthode est faite dans [75]. Les auteurs pointent quatre défauts majeurs de cette méthode :

- Les situations de blocages dues aux minima locaux (comme par exemple dans un U),

- Il peut ne pas y avoir de solution entre deux obstacles trop rapprochés alors même que le mouvement est possible,
- Il peut y avoir une oscillation en présence d’obstacle,
- Il peut y avoir une oscillation en présence de passage étroit.

Qui plus est, toute implémentation de nouveaux comportements pour le robot nécessite de proposer une expression mathématiques des potentiels parfois ardue à mettre en place. Ces approches ne sont donc pas très flexibles ni génériques, malgré tout elles restent simples à mettre en œuvre et donnent, dans les cas simples, des résultats satisfaisants.

1.5.5 Présentation du cadre de l’étude

Dans l’approche que nous allons détailler dans le chapitre 5, nous proposons une méthode qui combine les avantages des approches discrètes et ceux des approches continues. L’idée est d’assigner au bloc stratégie la partie discrète et au bloc planification de trajectoire la partie continue. Ainsi, à haut niveau, le bloc stratégie élabore un chemin global, fait de points de passage en utilisant une représentation discrète de l’environnement, tandis qu’à bas niveau le bloc de planification de trajectoire calcule une trajectoire continue permettant de relier les points de passage entre eux. L’avantage de cette méthode est de profiter de la puissance combinatoire des approches discrètes en l’alliant à la finesse d’analyse des approches continues. Cette finesse nous permet de prendre en compte, au niveau global, les différentes contraintes du système, notamment les contraintes cinématiques qui rendent, dans certains cas, impossible l’application sur un robot réel d’une trajectoire donnée (par exemple une trajectoire nécessitant des mouvements de translation pour un robot non-holonyme), ainsi que toutes les autres contraintes liées à la situation courante et à l’environnement (limitation de vitesse, non collision avec les obstacles ou les autres robots ...).

1.6 ORGANISATION DU MÉMOIRE DE THÈSE

1.6.1 Chapitre 2 - Boîte à outils

Le chapitre 2 présente la boîte à outils de ce mémoire. Elle se décompose en deux parties principales. La première partie concerne la modélisation des robots mobiles à roues. En s’appuyant sur la littérature nous étudierons les différents types de contraintes s’appliquant sur les roues d’un robot. Ces différentes contraintes liées au fait qu’il faut être capable de commander le robot permettent d’extraire cinq classes de robots mobiles pour lesquelles il est possible d’écrire un modèle cinématique générique. Une fois ces différentes classes présentées, la deuxième partie de la boîte à outils consistera à présenter les outils d’algèbre différentielle permettant l’étude de l’observabilité et de la platitude des systèmes. Un bref rappel sur les systèmes non linéaires sera fait avant de présenter l’utilisation des outils algébriques pour souligner un certain nombre de propriétés d’un système, notamment les propriétés de platitude. En guise d’exemple, cette étude sera menée sur les quatre des cinq modèles de robots présentés dans la première partie de ce chapitre. Finalement nous présenterons un dérivateur numérique en environnement bruité qui nous permettra, une fois les critères de localisabilité écrits (ces critères se basant sur des propriétés d’algèbre différentielle), de construire des estimateurs d’état pour les robots.

1.6.2 Chapitre 3 - Localisation d'un robot de type unicycle

Le chapitre 3 présente le premier cas d'étude de ce mémoire. Le cas du robot mobile unicycle (qui appartient à l'une des cinq classes présentées dans la boîte à outils, par exemple un robot ayant deux roues motrices parallèles et une roue folle est un robot unicycle) mesurant à l'aide d'une caméra les angles relatifs entre lui-même et des points d'intérêt de l'environnement. Il sera montré que l'outil « algèbre différentielle » facilite grandement l'étude d'un tel problème. Lorsque le robot ne mesure les angles relatifs que par rapport à un seul point d'intérêt (point de l'espace distinguable par le robot de coordonnées connues ou inconnues), à défaut de pouvoir reconstruire la position, il est possible de reconstruire les vitesses du robot. Par la suite avec deux points d'intérêt, quatre cas d'étude en fonction des hypothèses de mesures sont présentés. Dans chacun de ces cas, une étude de la localisabilité est effectuée avant de proposer un estimateur de position et de commandes (en fonction des cas). En guise de conclusion à ce chapitre, une étude en simulation et expérimentale est effectuée pour valider les algorithmes proposés en comparant leurs résultats à ceux donnés par un filtre de Kalman Étendu .

1.6.3 Chapitre 4 - Localisation dans un cadre général

Le chapitre 4 étend les résultats obtenus dans le chapitre 3 à l'ensemble des autres classes de robots. Grâce à l'étude menée sur les modèles cinématiques des robots dans la boîte à outils, les résultats pour les autres types de robots peuvent être facilement dérivés de ceux obtenus pour le robot unicycle. Comme dans le chapitre précédent, une étude en simulation vient valider le comportement des algorithmes.

1.6.4 Chapitre 5 - Navigation autonome d'une flottille de robots

Le chapitre 5 s'attarde sur la problématique de planification de trajectoire et de stratégie de navigation pour une flottille de robots unicycles. L'algorithme utilisé pour la planification de trajectoire (dérivé de travaux précédents des auteurs) est rappelé, les différentes modifications apportées dans ce mémoire sont ensuite présentées. Ces dernières consistent essentiellement à utiliser l'articulation entre le bloc stratégie et le bloc planification de trajectoire dans l'optique d'implémenter des comportements globaux plus complexes qu'auparavant. L'algorithme présenté ici est testé dans une expérimentation hautement dynamique et sur des cas de simulation représentatifs de cas réels. Ces différentes études valident le comportement attendu.

1.6.5 Chapitre 6 - Conclusion générale

Ce dernier chapitre donne une conclusion globale à ce mémoire en rappelant les principales contributions.

1.7 LE FAUTEUIL ROULANT ÉLECTRIQUE AUTONOME

Le travail présenté ici a été développé dans le cadre du projet SYSIASS, soutenu par le FEDER et le programme Interreg IVA 2 Mers. Ce projet a pour ambition de développer des modules d'aide à la conduite, de communication sécurisée de données et des interfaces homme-machine sans utilisation de main. Le module d'aide à la conduite, qui nous intéresse plus particulièrement dans ce travail, doit

être d'une part générique, c'est-à-dire adaptable à la plupart des fauteuils roulants électriques existant sur le marché, et d'autre part évolutif (capable de s'adapter aux besoins de l'utilisateur et/ou du corps médical, à l'état de l'utilisateur et à l'évolution de son handicap et capable de s'adapter et interagir avec son environnement). Il est important de notre point de vue, que le couple utilisateur-fauteuil garde un niveau d'autonomie constant. Trois modes de fonctionnement sont proposés pour ce module :

1. un mode manuel où l'utilisateur conduit le fauteuil et le module assure seulement sa sécurité, en lui indiquant par un retour visuel la direction et le degré de dangerosité des obstacles se trouvant sur sa trajectoire et en ralentissant la vitesse du fauteuil en conséquence,
2. un mode semi-automatique où l'utilisateur conduit le fauteuil et le module l'aide à contourner les obstacles se trouvant sur sa trajectoire,
3. un mode automatique où le module d'aide à la conduite prend à sa charge la conduite du fauteuil.

Le développement de ce projet se fait en étroite collaboration entre les scientifiques, le corps médical et les utilisateurs. Ce projet est encore en cours de développement aujourd'hui et l'ensemble des apports du travail présenté ici ont été effectués avec la volonté de pouvoir les implémenter sur un fauteuil roulant électrique. Nous avons donc effectué un certain nombre de choix technologiques liés à la plateforme finale, notamment le choix d'un capteur le moins cher possible et la nécessité de l'exécution de nos programmes en temps réel sur un ordinateur.

1.8 PUBLICATIONS

Le travail présenté ici a fait l'objet de trois publications dans des conférences internationales dont voici le rappel :

H. Sert, A. Kökösy, J. Palos, and W. Perruquetti. Mono landmark localization for an autonomous navigation of a cooperative mobile robot formation. *Second International Conference on Intelligent Robotics and Automation*, pages 1278–1292, Singapour, 16-18 Décembre, 2009.

H. Sert, A. Kokosy, and W. Perruquetti. A single landmark based localization algorithm for non-holonomic mobile robots. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 293–298, Shangai, Chine, May 2011.

H. Sert, W. Perruquetti, A. Kokosy, X. Jin, J. Palos. Localizability of unicycle mobiles robots : an algebraic point of view. In *2012 IEEE International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Algarve, Portugal, Octobre 2012.

BOÎTE À OUTILS

2

SOMMAIRE

2.1	MODÉLISATION DES ROBOTS MOBILES À ROUES	33
2.1.1	Définitions	33
2.1.2	Contraintes holonomes et non-holonomes	34
2.1.3	Commandabilité des systèmes non-holonomes	35
2.1.4	Classification des robots mobiles à roues	39
2.2	OBSERVABILITÉ, PLATITUDE ET ALGÈBRE DIFFÉRENTIELLE	51
2.2.1	Système et dynamiques	53
2.2.2	Étude des propriétés d'un système non-linéaire	55
2.2.3	Platitude dans le cas des robots mobiles à roues	59
2.2.4	Dérivation numérique	63
	CONCLUSION	69

DANS ce travail nous étudions deux problématiques principales de la robotique mobile : la planification de trajectoire et la localisation. Pour ce faire, nous nous appuyons sur la structure algébrique du système d'équations différentielles modélisant le robot, le modèle cinématique. Il existe cinq classes de robots mobiles non dégénérés [19]. Ainsi pour un robot mobile à roues donné, il est toujours possible, en fonction de son architecture mécanique, de le placer dans l'une de ces cinq classes. Dans chaque cas, il existe alors une représentation mathématique générique pour la classe considérée. Ainsi, une étude exhaustive est tout à fait possible puisqu'il suffit d'étudier pour chaque problématique (localisation, planification de trajectoire, . . .), les cinq modèles mathématiques. L'objectif de la première partie de ce chapitre sera donc de faire un bref rappel sur ce qu'est un robot mobile à roues et quelles sont les contraintes qui s'appliquent sur les roues d'un tel robot. C'est l'écriture générique de ces contraintes pour chaque type de roues qui donne la classification cinématique. Elle s'appuie sur les notions de degré de mobilité et de degré de directionnalité. Pour chaque type de robot nous donnerons son modèle cinématique et quelques détails sur l'architecture mécanique correspondante, ainsi qu'un exemple de robot représentatif de la classe.

Une fois ces différentes notions présentées, les outils utilisés pour la localisabilité et la planification de trajectoire seront présentés. A ces fins, nous utiliserons les techniques issues de l'analyse des systèmes non linéaire. Deux principales écoles existent pour l'étude de tels systèmes, une école dite « géométrique » et une école dite « algébrique ». Les méthodes algébro-différentielles présentent l'avantage de

caractériser le système et de proposer une solution directe à la problématique de localisation. Par exemple, pour l'observabilité d'un système non linéaire, la méthode géométrique conclura sur l'observabilité ou non du système (la synthèse de l'observateur se faisant alors dans un second temps), la méthode algébrique permettra en plus de proposer un estimateur d'état basé sur un différenciateur en environnement bruité que nous allons présenter dans la dernière partie de ce chapitre.

2.1 MODÉLISATION DES ROBOTS MOBILES À ROUES

Cette partie puise son inspiration dans les cours de robotique mobile de B. Bayle et de W. Perruquetti ainsi que dans [19].

2.1.1 Définitions

Les robots mobiles étudiés dans ce travail sont des véhicules à roues capables de se mouvoir de façon autonome (c'est-à-dire sans l'aide d'un opérateur humain). Nous supposons ici que ces véhicules sont conçus à partir d'éléments rigides et équipés de roues indéformables évoluant dans un plan horizontal. La figure 2.1-(a) donne un exemple d'un tel robot, le fauteuil roulant électrique autonome. La position du véhicule dans le plan est décrite par trois variables (x, y, θ) comme le montre la figure 2.1-(b).

On note $\mathcal{R} = (O, \vec{x}, \vec{y}, \vec{z})$ un repère fixe quelconque, dont l'axe z est vertical et $\mathcal{R}' = (O', \vec{x}', \vec{y}', \vec{z}')$ un repère mobile lié au robot (les axes \vec{z} et \vec{z}' sont différents). On choisit généralement pour O' un point remarquable de la plateforme, typiquement le centre de l'axe des roues motrices s'il existe.

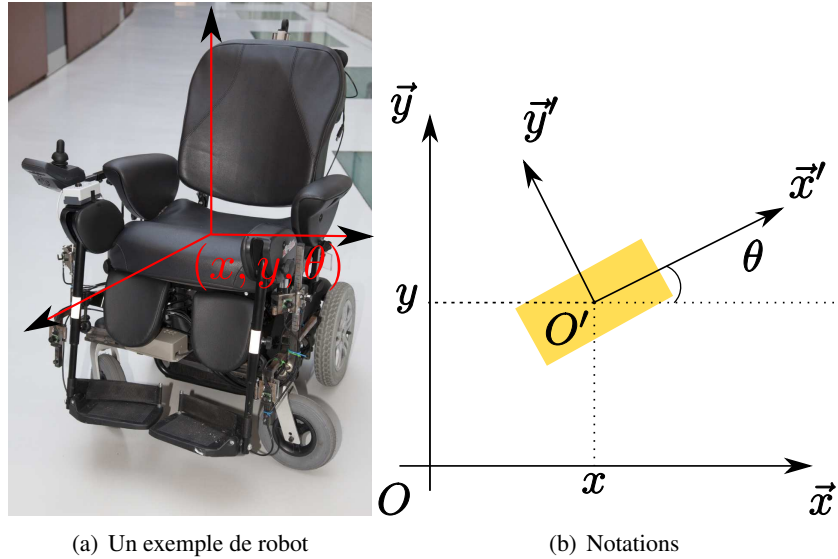


FIGURE 2.1 – Notation utilisée et premier exemple de robot mobile

Nous pouvons alors définir les deux notions suivantes :

Définition 2.1 (Posture) [19] La **posture** \mathbf{p} d'un robot est le vecteur :

$$\mathbf{p} = (x, y, \theta)^T,$$

où x et y sont respectivement l'abscisse et l'ordonnée du point O' dans \mathcal{R} et θ l'angle (\vec{x}, \vec{x}') (voir la figure 2.1-b). La **posture** du robot est donc définie dans un espace \mathcal{M} de dimension $m = 3$, comparable à l'espace opérationnel d'un manipulateur plan.

Définition 2.2 (Configuration) [108] La **configuration** d'un système dynamique est connue quand la position de tous ses points dans un repère donné est connue. Cette configuration est représentée comme un vecteur

$$\mathbf{q} = (\mathbf{q}_1, \dots, \mathbf{q}_n)^T,$$

de n coordonnées appelées coordonnées généralisées. La **configuration** est ainsi définie sur un espace \mathcal{N} de dimension n , appelée l'espace des configurations.

- Exemple 2.1**
- Pour un bras manipulateur, cette notion de configuration est définie sans ambiguïté par les positions angulaires des différentes articulations,
 - Pour une roue, la configuration est la position du centre de la roue dans le plan de mouvement, son orientation ainsi que son angle de rotation.

2.1.2 Contraintes holonomes et non-holonomes

Un système mécanique peut être représenté sous la forme d'un ensemble de contraintes de position et/ou de vitesse entre les différents points du système. On distingue deux principaux types de contraintes :

- **Contraintes holonomes** : Ce sont les contraintes intégrables. Elles génèrent des relations algébriques liant les différentes coordonnées de position,
- **Contraintes non holonomes** : Ce sont les contraintes non intégrables. Elles génèrent des relations algébriques dont on ne peut éliminer les vitesses de certains points caractéristiques.

- Exemple 2.2** – Deux exemples de contraintes :

$$a\dot{x}_1 = b\dot{x}_2 \quad (2.1)$$

est une contrainte holonome (en intégrant on a $a(x_1(t) - x_1(0)) = b(x_2(t) - x_2(0))$)

$$\dot{x}_1 \sin(x_3) = \dot{x}_2 \cos(x_3) \text{ est une contrainte non holonome}$$

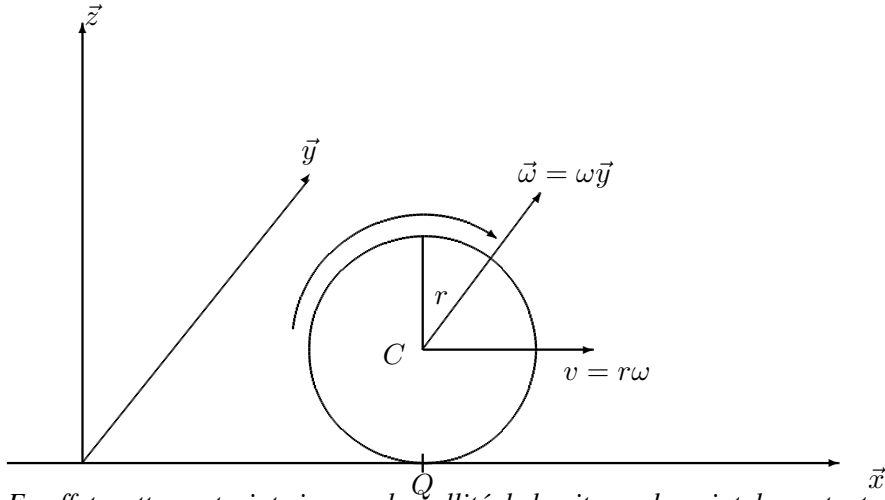
- Prenons un corps rigide. La rigidité de ce corps est un ensemble de contraintes holonomes scléronomes (c'est-à-dire qu'elles ne dépendent pas du temps). Pour deux points quelconques numérotés i, j il existe une constante $C_{i,j}$ telle que l'on doit avoir $\|\vec{r}_i - \vec{r}_j\| = C_{i,j}$,

Dans le cadre d'un système holonome le retour à l'origine des variables articulaires signifie un retour à la position d'origine. Par contre, pour un système non-holonome, le retour à l'origine de la configuration des roues ne garantit pas le retour du système à la position d'origine (le mouvement d'un système non holonome dépend du chemin suivi). Nous étudierons les contraintes sur la roue plus en détail dans la suite de ce manuscrit.

Les contraintes de non-holonomie d'un robot mobile proviennent de l'expression mathématique des conditions de contact entre les roues et le sol. Elles donnent des contraintes de vitesses non intégrables.

Définition 2.3 (Contrainte de roulement sans glissement) *En supposant que le contact entre la roue et le sol est ponctuel et que la roue est indéformable, la contrainte de **roulement sans glissement** est vérifiée, c'est-à-dire que la vitesse du point de contact entre la roue et le sol est nulle.*

- Exemple 2.3** La contrainte de roulement sans glissement nous permet d'écrire la vitesse de translation d'une roue en mouvement de rotation :



En effet, cette contrainte impose la nullité de la vitesse du point de contact entre la roue et le sol donc :

$$\vec{v}_Q = \vec{0} = \vec{v}_C + \omega \wedge \overrightarrow{CQ} \quad (2.2)$$

Or $Q = (x, 0, 0)^T$, $C = (x, 0, r)^T$, donc $\overrightarrow{CQ} = (0, 0, -r)$, de plus $\vec{\omega} = (0, \omega, 0)^T$ et donc :

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{x} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \omega \\ 0 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ 0 \\ -r \end{pmatrix} \quad (2.3)$$

Donc

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{x} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -r\omega \\ 0 \\ 0 \end{pmatrix} \quad (2.4)$$

Et finalement :

$$\dot{x} = r\omega \quad (2.5)$$

2.1.3 Commandabilité des systèmes non-holonomes

Définition 2.4 (Contrainte de non holonomie) *Les contraintes de non holonomie en robotique et en mécanique sont définies par des conditions cinématiques qui ne peuvent pas être représentées par une équation $\mathbf{f}(\mathbf{q}, t) = 0$ contenant seulement les coordonnées généralisées \mathbf{q} et le temps t .*

Pour les robots mobiles à roues, ces contraintes sont formulées par des équations différentielles non intégrables du premier ordre, contenant les dérivées temporelles des coordonnées généralisées, et représentées sous la forme Pfaffienne :

$$H^T(\mathbf{q})\dot{\mathbf{q}} = 0, \quad (2.6)$$

L'existence de contraintes non-holonomes implique que le système ne peut pas effectuer certains mouvements instantanément. Par exemple, dans le cas de la roue, il ne peut y avoir de translation instantanée parallèlement à l'axe de rotation de la roue. Un tel déplacement nécessitera des manœuvres. De même, comme on le sait bien, une voiture ne peut se garer facilement sans effectuer de créneaux.

Définition 2.5 (Degré de liberté, (d.l)) *Le degré de liberté (d.l) est défini comme la différence entre le nombre de coordonnées généralisées n et le nombre de contraintes indépendantes m :*

$$d.l = n - m.$$

Écrivons l'équation d'état comme dans le cas des systèmes sans termes de dérive sous la forme :

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})\mathbf{u}, \mathbf{x} \in \mathbb{R}^n, \quad (2.7)$$

où $\mathbf{f} = (\mathbf{f}_1(x), \mathbf{f}_2(x), \dots, \mathbf{f}_{n-m}(x))^T$ est constituée de champs de vecteurs libres ($\text{rang}(\mathbf{f}) = n - m$) et $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n-m})^T$ vecteur des entrées de commande. Notons que le rang de l'algèbre de Lie (dite aussi de contrôlabilité) générée par les crochets de Lie de $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{n-m}$ est n . La notion de **crochet de Lie** est expliquée en détail dans l'annexe A.3.

Définition 2.6 (Algèbre de Lie) Soient $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_p$ des champs de vecteurs que l'on suppose libres (indépendants les uns des autres). **L'algèbre de Lie** engendrée par les champs de vecteurs \mathbf{f}_i est la distribution construite à partir de $\text{vect}\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_p\}$ à laquelle on ajoute tous les crochets de Lie successifs formés à partir des \mathbf{f}_i à condition qu'ils augmentent la dimension de l'algèbre (c'est-à-dire qui ne sont pas déjà engendrés par la distribution en cours de construction). On la note $\mathcal{A}\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_p\}$.

La matrice constituée des champs de vecteurs qui engendrent \mathcal{A} permet de caractériser l'accessibilité (contrôlabilité) d'un système donné. Ainsi, si $\text{rang}(\mathcal{A}\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_p\}) = n$ alors le système $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})\mathbf{u}, \mathbf{x} \in \mathbb{R}^n, \mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_p)$ est accessible (localement commandable).

Prenons un système ayant pour représentation mathématique le système d'équations suivant (nous verrons plus tard que ce sont les équations d'un robot unicycle, les notations sont données sur la figure 2.2)

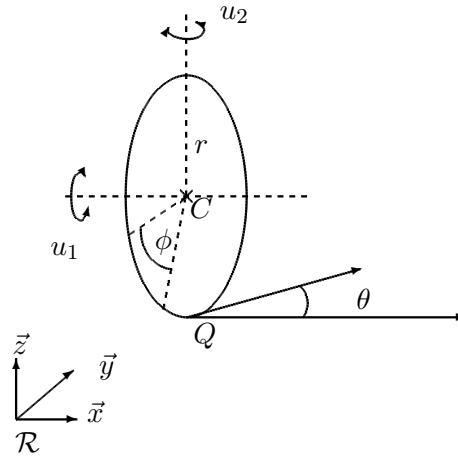


FIGURE 2.2 – Notation pour l'étude d'une roue

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2 \quad (2.8)$$

$$f_1 = (\cos(\theta), \sin(\theta), 0)^T, \quad (2.9)$$

$$f_2 = (0, 0, 1)^T, \quad (2.10)$$

$$[f_1, f_2] = (-\sin(\theta), \cos(\theta), 0)^T, \quad (2.11)$$

$$([f_1, g_2], f_1, f_2) = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & 0 \\ \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\dim(\text{vect}\{[f_1, f_2], f_1, f_2\}) = 3,$$

Un tel système est donc localement commandable.

Il n'est pas évident de dire *a priori* si une contrainte est intégrable ou non. Pour étudier une contrainte donnée, il est possible d'utiliser le théorème de Frobenius (voir [153] ou [112]). Pour appliquer ce théorème, seul le **crochet de Lie** est nécessaire.

Théorème 2.1 *Soit un système de configuration \mathbf{q} , de dimension n , soumis à un ensemble de contraintes indépendantes s'écrivant sous la forme $H^T(\mathbf{q})\dot{\mathbf{q}} = 0$ (Équation (2.6)) Soit $H^\perp(\mathbf{q}) = (h_1^\perp(\mathbf{q}), h_2^\perp(\mathbf{q}), \dots, h_m^\perp(\mathbf{q}))$ une matrice de rang plein m , orthogonale à $H(\mathbf{q})$ sur tout l'espace de configurations. Soit enfin l'algèbre de Lie $\mathcal{A}(H^\perp)$ de dimension p , avec $m \leq p \leq n$. Alors, parmi les contraintes auxquelles est soumis le système, $n-p$ sont intégrables et sont donc des contraintes holonomes.*

Exemple 2.4 (Calcul du nombre de contraintes non-holonomes d'une roue) *Prenons une roue de rayon r satisfaisant la contrainte de roulement sans glissement. Soit C le centre de la roue de coordonnées (x, y, r) et Q le point de contact de coordonnées $(x, y, 0)$ dans le repère \mathcal{R} précédemment défini. Notons ϕ l'angle de rotation propre de la roue et θ l'orientation de la roue par rapport au repère \mathcal{R} . Soit \vec{u}_2 la vitesse de rotation de la roue, \vec{v}_C la vitesse du point C et \vec{v}_Q la vitesse du point Q dans le repère \mathcal{R} (voir les notations sur la figure 2.2). La contrainte de roulement sans glissement s'écrit :*

$$\vec{v}_Q = \vec{v}_C + \vec{u}_2 \wedge \overrightarrow{CQ} = \vec{0}. \quad (2.12)$$

Ce qui donne :

$$\dot{x}\vec{x} + \dot{y}\vec{y} + \left(\dot{\theta}\vec{z} + \dot{\phi}(\sin\theta\vec{x} - \cos\theta\vec{y}) \right) \wedge (-r\vec{z}) = \vec{0}, \quad (2.13)$$

$$(\dot{x} + r\dot{\phi}\cos\theta)\vec{x} + (\dot{y} + r\dot{\phi}\sin\theta)\vec{y} = \vec{0}. \quad (2.14)$$

On obtient alors le système de contraintes suivant :

$$\dot{x} + r\dot{\phi}\cos\theta = 0, \quad (2.15)$$

$$\dot{y} + r\dot{\phi}\sin\theta = 0. \quad (2.16)$$

Et finalement :

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0, \quad (2.17)$$

$$\dot{x}\cos\theta + \dot{y}\sin\theta = -r\dot{\phi}. \quad (2.18)$$

Notons que de simples manipulations conduisent à :

$$\dot{x} = \cos(\theta)(-r\dot{\phi}), \quad (2.19)$$

$$\dot{y} = \sin(\theta)(-r\dot{\phi}). \quad (2.20)$$

Et donc en posant $u_2 = \dot{\theta}$ et $u_1 = -r\dot{\phi}$, on retrouve le système (2.8) Notons sa configuration $\mathbf{q} = (x, y, \theta, \phi)^T$. Ces contraintes se mettent sous la forme (2.6) i.e. $H(\mathbf{q})\dot{\mathbf{q}} = 0$, avec :

$$H^T(\mathbf{q})\dot{\mathbf{q}} = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & 0 & 0 \\ \cos(\theta) & \sin(\theta) & 0 & r \end{pmatrix} \times \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = 0.$$

Nous cherchons maintenant une matrice $H^\perp(\mathbf{q})$ qui soit orthogonale à $H^T(\mathbf{q})$ sur tout l'espace de configurations, c'est-à-dire que $H^T(\mathbf{q}) \times H^\perp(\mathbf{q}) = 0$:

$$H^\perp(\mathbf{q}) = \begin{pmatrix} h_{11}^\perp(\mathbf{q}) & h_{12}^\perp(\mathbf{q}) \\ h_{21}^\perp(\mathbf{q}) & h_{22}^\perp(\mathbf{q}) \\ h_{31}^\perp(\mathbf{q}) & h_{32}^\perp(\mathbf{q}) \\ h_{41}^\perp(\mathbf{q}) & h_{42}^\perp(\mathbf{q}) \end{pmatrix}.$$

$$-\sin(\theta)h_{11}^\perp(\mathbf{q}) + \cos(\theta)h_{21}^\perp(\mathbf{q}) = 0, \quad (2.21)$$

$$\cos(\theta)h_{11}^\perp(\mathbf{q}) + \sin(\theta)h_{21}^\perp(\mathbf{q}) + rh_{41}^\perp(\mathbf{q}) = 0, \quad (2.22)$$

$$-\sin(\theta)h_{12}^\perp(\mathbf{q}) + \cos(\theta)h_{22}^\perp(\mathbf{q}) = 0, \quad (2.23)$$

$$\cos(\theta)h_{12}^\perp(\mathbf{q}) + \sin(\theta)h_{22}^\perp(\mathbf{q}) + rh_{42}^\perp(\mathbf{q}) = 0. \quad (2.24)$$

Alors :

$$H^\perp(\mathbf{q}) = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \\ -\frac{1}{r} & 0 \end{pmatrix}.$$

En utilisant $h_1^\perp(\mathbf{q}) = (\cos(\theta) \ \sin(\theta) \ 0 \ -\frac{1}{r})^T$, $h_2^\perp(\mathbf{q}) = (0 \ 0 \ 1 \ 0)^T$, calculons les crochets de Lie successifs

$$\begin{aligned} [h_1^\perp, h_2^\perp](\mathbf{q}) &= \left(\frac{\partial h_2^\perp}{\partial \mathbf{q}} h_1^\perp - \frac{\partial h_1^\perp}{\partial \mathbf{q}} h_2^\perp \right)(\mathbf{q}) = 0 - \begin{pmatrix} 0 & 0 & -\sin(\theta) & 0 \\ 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \\ &= \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \quad (2.25)$$

$$\begin{aligned}
[h_1^\perp, [h_1^\perp, h_2^\perp]](\mathbf{q}) &= \\
\begin{pmatrix} 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & \sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \\ -\frac{1}{r} \end{pmatrix} &- \begin{pmatrix} 0 & 0 & -\sin(\theta) & 0 \\ 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \\ 0 \end{pmatrix}, \\
&= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.
\end{aligned} \tag{2.26}$$

$$\begin{aligned}
[h_2^\perp, [h_1^\perp, h_2^\perp]](\mathbf{q}) &= \begin{pmatrix} 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & \sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \\
&= \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \\ 0 \end{pmatrix}.
\end{aligned} \tag{2.27}$$

$$\begin{aligned}
D &= \det \begin{pmatrix} \cos(\theta) & \sin(\theta) & \cos(\theta) & 0 \\ \sin(\theta) & -\cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{1}{r} & 0 & 0 & 0 \end{pmatrix}, \\
&= -\det \begin{pmatrix} \cos(\theta) & \sin(\theta) & \cos(\theta) \\ \sin(\theta) & -\cos(\theta) & \sin(\theta) \\ -\frac{1}{r} & 0 & 0 \end{pmatrix}, \\
&= \frac{1}{r} \det \begin{pmatrix} \sin(\theta) & \cos(\theta) \\ -\cos(\theta) & \sin(\theta) \end{pmatrix}, \\
&= \frac{1}{r} \neq 0.
\end{aligned} \tag{2.28}$$

Comme $\mathcal{A}(H^\perp)$ est de dimension 4, nous pouvons retirer $4 - 4 = 0$ contrainte donc les deux contraintes précédentes sont toutes les deux non-holonomes.

2.1.4 Classification des robots mobiles à roues

2.1.4.1 Hypothèse de travail

Nous considérons un robot équipé de N roues : les roues et le véhicule sont indéformables (le robot n'est pas équipé de suspension). Durant le mouvement (supposé plan), nous supposons que le plan de mouvement de chacune des roues reste vertical¹ et que son contact avec le sol est réduit à un seul point du plan.

1. l'axe de direction des roues est perpendiculaire au sol et l'axe de traction des roues est parallèle au sol.

L'orientation de la roue par rapport au robot peut par contre être fixe ou variable. Deux types de roues peuvent satisfaire ces différentes hypothèses : le premier, les roues *conventionnelles* pour lesquelles le contact entre la roue et le sol satisfait la condition de roulement parfait et sans glissement (voir la définition 2.3), ce qui implique que la vitesse du point de contact est nulle. Le deuxième type de roues, ce sont les roues *suédoises* pour lesquelles seule une composante de la vitesse du point de contact de la roue avec le sol est supposée nulle et ceci tout le long du mouvement. La direction de cette composante nulle est *a priori* arbitraire, mais elle est déterminée par son angle γ par rapport à l'orientation de la roue.

2.1.4.2 Notations utilisées

La figure 2.3 présente les six paramètres utilisés pour définir la position d'une roue dans son environnement. Ces six paramètres sont les trois angles ($\alpha \beta \gamma$) et les trois longueurs ($l d r$).

Nous notons $\mathcal{R}_{global} = (O, \vec{x}, \vec{y})$ le référentiel global, $\mathcal{R}_{robot} = (P, \vec{x}_2, \vec{y}_2)$ le repère lié au robot et $\mathcal{R}_{roue} = (Q, \vec{x}_3, \vec{y}_3)$ le repère lié à la roue. Nous notons θ l'angle de rotation entre \mathcal{R}_{global} et \mathcal{R}_{robot} et $\alpha + \beta$ l'angle de rotation entre \mathcal{R}_{robot} et \mathcal{R}_{roue} . β dépend du temps pour les roues orientables centrées et décentrées et est fixe pour les roues fixes. L'angle de rotation de la roue autour de son axe (horizontal) est décrite par l'angle variable $\phi(t)$. Le rayon de la roue est noté r . Nous notons respectivement z_O, z_P, z_Q, z_A l'affixe des points O, P, Q, A dans le référentiel \mathcal{R}_{global} . Quand il y a ambiguïté sur le repère utilisé, il est indiqué en indice de la variable utilisée. En coordonnées cartésiennes, nous notons \mathbf{p} les coordonnées du point P tel que $\mathbf{p} = (x, y, \theta)$. Le point A est tel que $PA = l$.

2.1.4.3 Les différents types de roues

A partir des six paramètres précédemment introduits, trois types de roues conventionnelles (qui satisfont la contrainte de roulement sans glissement) et un type de roue suédoise peuvent être définies (cf. figure 2.4) :

1. Les **roues conventionnelles centrées fixes** pour lesquelles l est nulle et β est une constante : l'axe de rotation, parallèle au sol, passe par le centre de la roue.
2. Les **roues conventionnelles orientables centrées** pour lesquelles l est nulle mais $\beta(t)$ varie en fonction du temps : l'axe d'orientation, perpendiculaire au sol, passe par le centre de la roue,
3. Les **roues conventionnelles orientables décentrées** pour lesquelles l est non nulle et $\beta(t)$ dépend du temps : l'axe d'orientation, perpendiculaire au sol, ne passe pas par le centre de la roue,
4. Les **roues suédoises** pour lesquelles β est constant et $\gamma \neq 0$: la composante nulle de la vitesse de glissement du point de contact n'est pas dans le plan de la roue.

Nous allons dans la partie suivante écrire les contraintes qui s'appliquent sur les roues. Nous allons effectuer une étude générique avant d'écrire pour chaque type de roue les contraintes spécifiques.

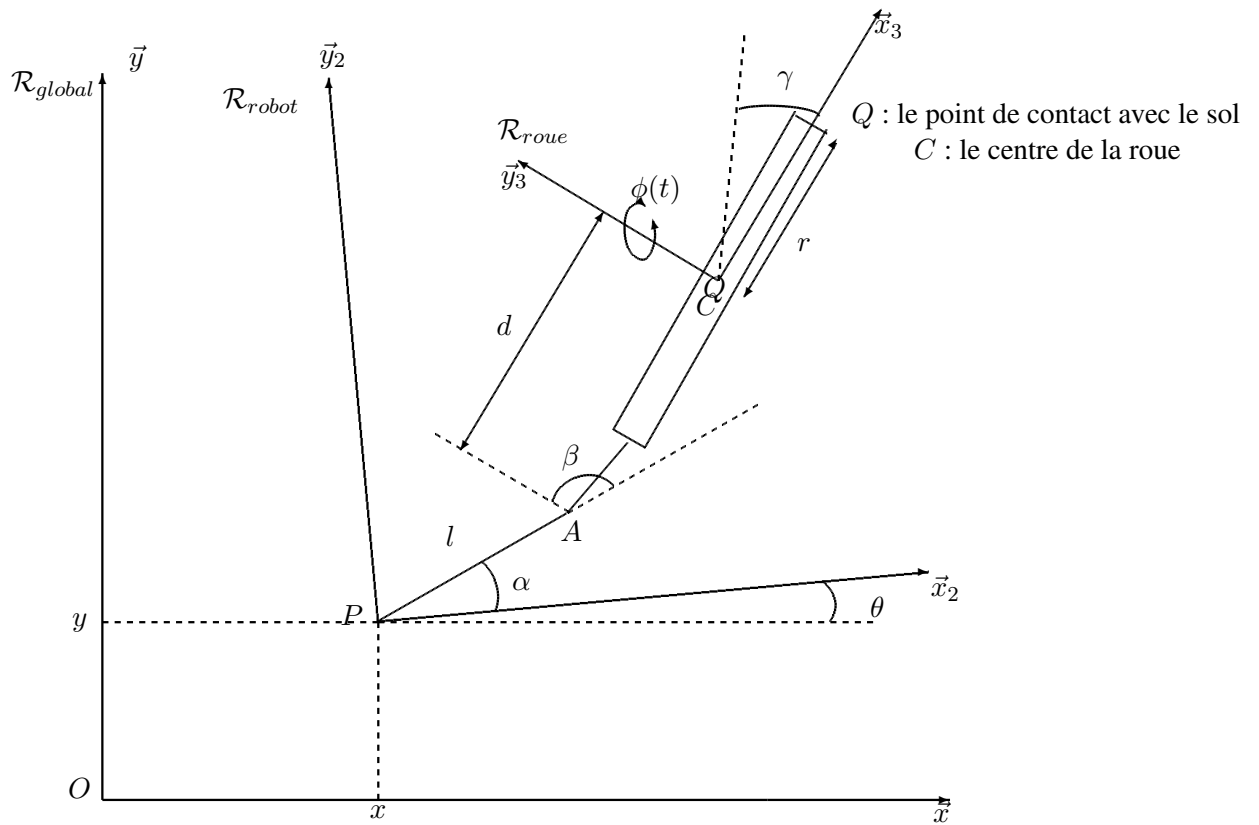


FIGURE 2.3 – Conventions de notations pour les roues

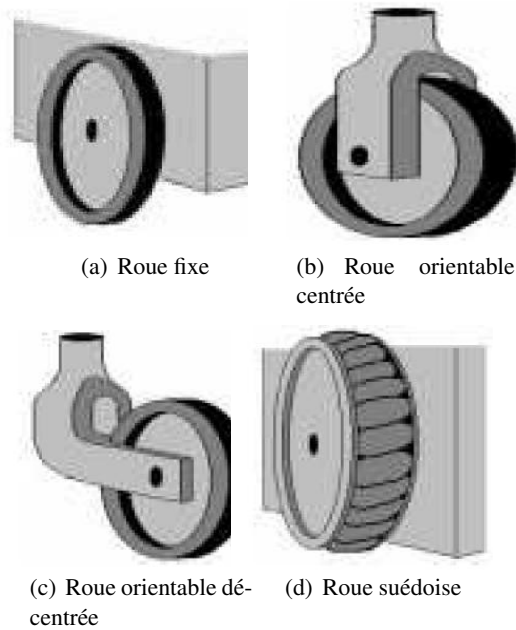


FIGURE 2.4 – Différents types de roues utilisées pour les plates-formes mobiles.

2.1.4.4 Écriture des contraintes

La contrainte de roulement sans glissement impose que la vitesse du point de contact entre la roue et le sol soit nulle. Notons Q le point de contact entre la roue

et le sol, et C le centre de la roue. La contrainte de roulement sans glissement dans le repère global s'écrit (où i est la variable complexe telle que $i^2 = -1$:

$$v_Q = v_C + ir\dot{\phi}(t) \cos(\gamma) + l\dot{\beta} = 0$$

Cherchons donc la vitesse du point C . L'affixe du point C noté z_C dans le repère \mathcal{R}_{global} est telle que :

$$z_C = z_P + l \exp(i(\alpha + \theta))$$

En dérivant par rapport au temps :

$$v_{C/\mathcal{R}_{global}} = \dot{z}_P + i\dot{\theta}l \exp(i(\alpha + \theta))$$

A partir de cette formule, il est possible de retrouver la vitesse du point C dans le référentiel \mathcal{R}_{roue} en effectuant les deux rotations permettant de passer de \mathcal{R}_{global} à \mathcal{R}_{roue} :

$$v_{C/\mathcal{R}_{roue}} = \exp(-i\theta) \times \exp(-i(\alpha + \beta + \gamma)) \times (\dot{z}_P + i\dot{\theta}l \exp(i(\alpha + \theta)))$$

Ce qui donne après simplification :

$$v_{C/\mathcal{R}_{global}} = \dot{z}_P \exp(-i(\alpha + \theta + \beta + \gamma)) + i\dot{\theta}l \exp(-i\beta)$$

Et finalement la contrainte de roulement sans glissement s'écrit :

$$v_Q = \dot{z}_P \exp(-i(\alpha + \theta + \beta + \gamma)) + i\dot{\theta}l \exp(-i\beta) + ir\dot{\phi}(t) + d\dot{\beta} = 0$$

Nous pouvons maintenant, en séparant partie réelle et partie imaginaire écrire les deux contraintes suivantes :

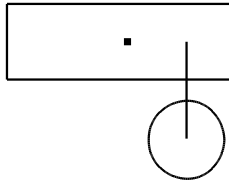
$$\dot{x} \cos(\alpha + \theta + \beta) + \dot{y} \sin(\alpha + \theta + \beta) + \dot{\theta}l \sin(\beta) + d\dot{\beta} = 0 \quad (2.29)$$

$$\dot{y} \cos(\alpha + \theta + \beta) - \dot{x} \sin(\alpha + \theta + \beta) + \dot{\theta}l \cos(\beta) + r\dot{\phi}(t) \cos(\gamma) = 0 \quad (2.30)$$

Remarque 2.1 Pour une roue suédoise, seule la composante imaginaire de la vitesse v_Q est nulle, ce qui donne uniquement la contrainte suivante :

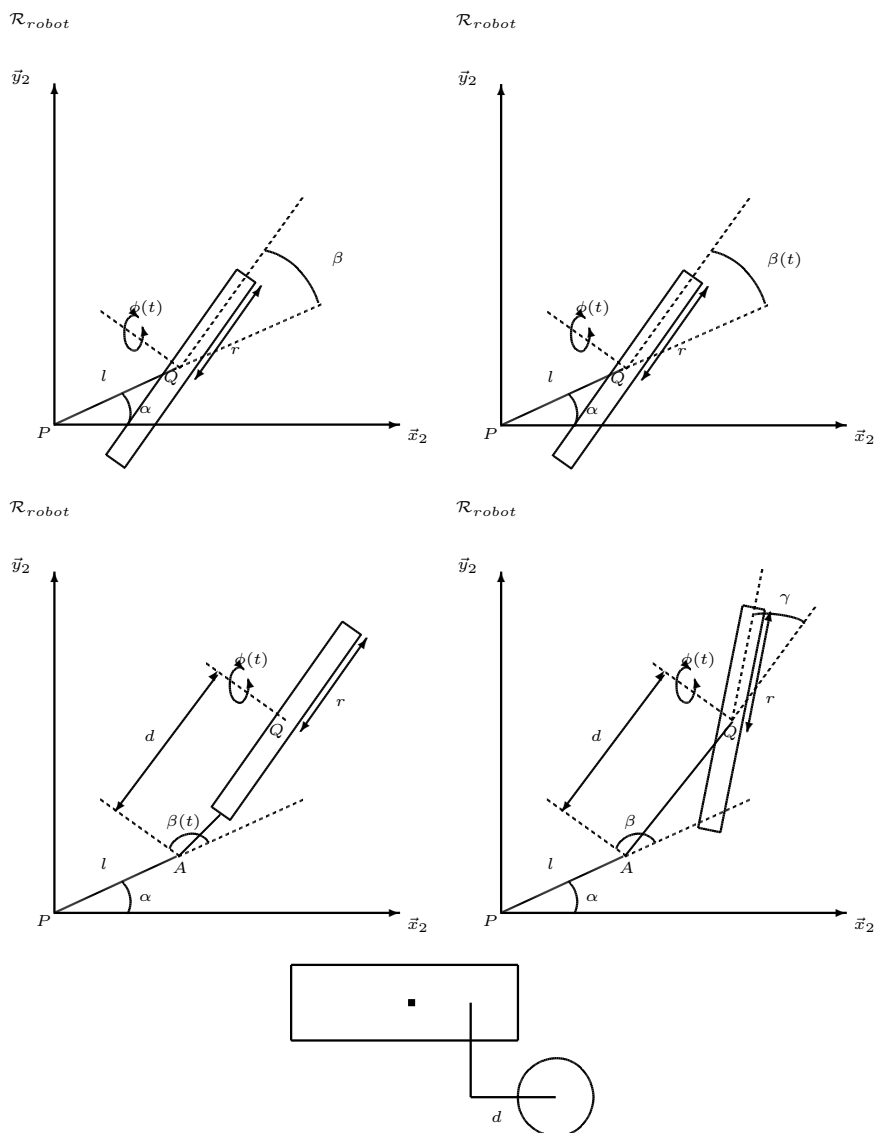
$$\dot{y} \cos(\alpha + \theta + \beta) - \dot{x} \sin(\alpha + \theta + \beta) + \dot{\theta}l \cos(\beta) + r\dot{\phi}(t) \cos(\gamma) = 0 \quad (2.31)$$

Type de roue	Valeurs des paramètres	Contraintes
conventionnelle centrée fixe	$d = 0$ $\dot{\beta} = 0$ $\gamma = 0$	$\dot{x} \cos(\alpha + \theta + \beta) + \dot{y} \sin(\alpha + \theta + \beta) + \dot{\theta}l \sin(\beta) = 0$ $\dot{y} \cos(\alpha + \theta + \beta) - \dot{x} \sin(\alpha + \theta + \beta) + \dot{\theta}l \cos(\beta) + r\dot{\phi}(t) = 0$
conventionnelle orientable centrée	$d = 0$ $\dot{\beta} \neq 0$ $\gamma = 0$	$\dot{x} \cos(\alpha + \theta + \beta) + \dot{y} \sin(\alpha + \theta + \beta) + \dot{\theta}l \sin(\beta) = 0$ $\dot{y} \cos(\alpha + \theta + \beta) - \dot{x} \sin(\alpha + \theta + \beta) + \dot{\theta}l \cos(\beta) + r\dot{\phi}(t) = 0$
conventionnelle orientable décentrée	$d \neq 0$ $\dot{\beta} \neq 0$ $\gamma = 0$	$\dot{x} \cos(\alpha + \theta + \beta) + \dot{y} \sin(\alpha + \theta + \beta) + \dot{\theta}l \sin(\beta) + d\dot{\beta} = 0$ $\dot{y} \cos(\alpha + \theta + \beta) - \dot{x} \sin(\alpha + \theta + \beta) + \dot{\theta}l \cos(\beta) + r\dot{\phi}(t) = 0$
suédoise	$\dot{\beta} = 0$ $\gamma \neq 0$	$\dot{y} \times \cos(\alpha + \theta + \beta) - \dot{x} \sin(\alpha + \theta + \beta) + \dot{\theta}l \cos(\beta) + r\dot{\phi}(t) \cos(\gamma) = 0$



Type de roue	(a)	(b)	(c)	(d)
r	constant	constant	constant	constant
b	constant	constant	constant	constant
φ	variable	variable	variable	variable
β	constant	variable	variable	constant
α	constant	constant	constant	constant
γ	0	0	0	constant
b'	0	0	0	constant

TABLE 2.1 – Propriétés des paramètres de description des roues en fonction de leur type



2.1.4.5 Degré de mobilité et centre instantané de rotation

Considérant maintenant, un robot mobile quelconque, équipé de N roues (chacune des roues étant de l'un des quatre types présentés auparavant). Dans ce cas :

$$N = N_{cf} + N_{coc} + N_{cod} + N_s \quad (2.32)$$

avec

- N_{cf} : nombre de roues conventionnelles fixes,
- N_{coc} : nombre de roues conventionnelles orientables centrées,
- N_{cod} : nombre de roues conventionnelles orientables décentrées,
- N_s : nombre de roues suédoises.

La configuration du robot est alors complètement décrite par les vecteurs suivants :

- Coordonnées de position (posture) $\mathbf{p}(t)$: pour la position et l'orientation dans le plan,
- Coordonnées angulaires $\beta_{coc}(t)$: pour les angles d'orientation des roues orientables centrées et $\beta_{cod}(t)$ pour les roues orientables décentrées,
- Coordonnées de rotation $\phi(t)$: pour l'angle d'orientation des roues autour des axes horizontaux.

$(\mathbf{p}(t), \beta_{coc}(t), \beta_{cod}, \phi(t))^T$ constituent les coordonnées de configuration : le nombre total étant $3 + N_{cf} + 2N_{coc} + 2N_{cod} + N_s$. En rassemblant l'ensemble des contraintes nous obtenons :

$$J_1(\beta_{coc}, \beta_{cod})R(\theta)\dot{\mathbf{p}} + J_2\dot{\phi} = 0 \quad (2.33)$$

$$C_1(\beta_{coc}, \beta_{cod})R(\theta)\dot{\mathbf{p}} + C_2\dot{\beta} = 0 \quad (2.34)$$

où

$$J_1(\beta_{coc}, \beta_{cod}) = \begin{pmatrix} J_{1cf} \\ J_{1coc}(\beta_{coc}) \\ J_{1cod}(\beta_{cod}) \\ J_{1s} \end{pmatrix}, J_2 = \text{diag}(r_i) \quad (2.35)$$

$$C_1(\beta_{coc}, \beta_{cod}) = \begin{pmatrix} C_{1cf} \\ C_{1coc}(\beta_{coc}) \\ C_{1cod}(\beta_{cod}) \end{pmatrix}, C_2 = \begin{pmatrix} 0 \\ 0 \\ C_{2cod} \end{pmatrix} \quad (2.36)$$

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}. \quad (2.37)$$

(2.34) donne en ne gardant que les termes correspondant aux roues **cf** et **coc** :

$$C_{1cf}R(\theta)\dot{\mathbf{p}} = 0 \quad (2.38)$$

$$C_{1coc}(\beta_{coc})R(\theta)\dot{\mathbf{p}} = 0 \quad (2.39)$$

Notons que dans ce cas $R(\theta)\dot{\mathbf{p}}$ appartient au noyau de la matrice :

$$C_1^*(\beta_c) = \begin{pmatrix} C_{1cf} \\ C_{1coc}(\beta_{coc}) \end{pmatrix}, \quad (2.40)$$

i.e. $R(\theta)\dot{\mathbf{p}} \in \text{Ker}(C_1^*(\beta_{coc}))$ avec $C_{1cf}, C_{1coc}(\beta_{coc})$ des matrices respectivement de dimension $(N_{cf} \times 3), (N_{coc} \times 3)$ et dont les lignes sont dérivées des contraintes.

Proposition 2.1 Soit v_i une base de $\text{Ker}(C_1^*(\beta_{coc}))$, alors

$$\dot{\mathbf{p}} = R^T(\theta) \sum_{i=1}^{\dim(\text{Ker}(C_1^*(\beta_{coc})))} a_i(t)v_i$$

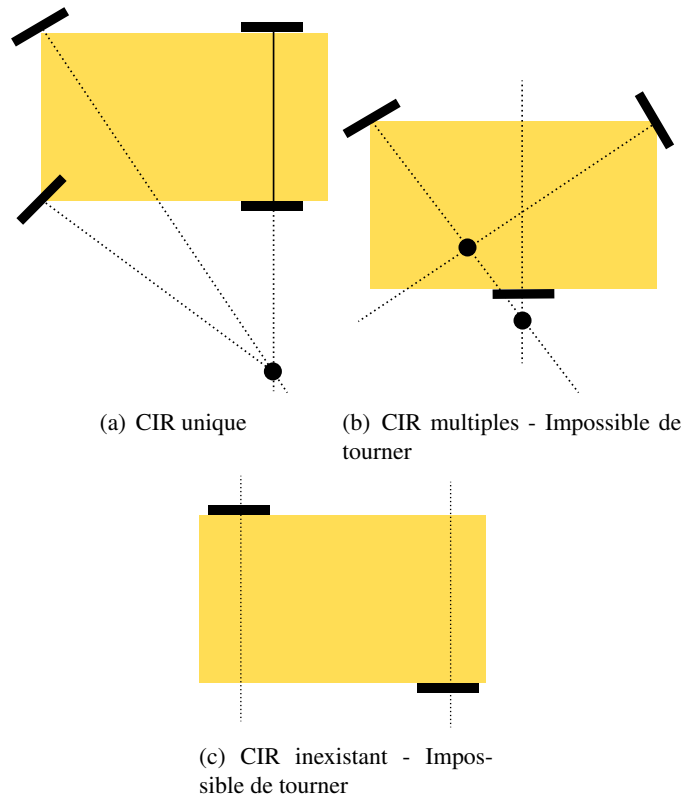


FIGURE 2.5 – Exemples et contre-exemple de centre instantané de rotation (CIR)

Nécessairement on a $\text{rang}(C_1^*(\beta_{coc})) \leq 3$. D'autre part si $\text{rang}(C_1^*(\beta_{coc})) = 3$ alors $\text{Ker}(C_1^*(\beta_{coc}))$ est réduit à 0 et donc $R(\theta)\dot{\mathbf{p}} = 0$: il ne peut y avoir de mouvement.

On voit que la non-holonomie des contraintes cinématiques impose des restrictions sur la mobilité du robot. La description de l'orientation des roues est très discutable. Parmi toutes les configurations possibles, seulement quelques-unes permettent la mobilité du robot en satisfaisant le roulement pur et le non glissement.

Bien évidemment, pour un ensemble de roues donné, toute disposition ne conduit pas à une solution viable. Un mauvais choix peut limiter la mobilité du robot ou occasionner d'éventuels blocages. Par exemple, un robot équipé de deux roues fixes non parallèles ne pourrait pas aller en ligne droite.

Définition 2.7 (Centre instantané de rotation) *On appelle **centre instantané de rotation (CIR)** le point de vitesse nulle lié aux roues autour duquel tourne le robot de façon instantanée.*

Proposition 2.2 *Les points de vitesse nulle liés aux roues se trouvant sur leur axe de rotation, le CIR est le point d'intersection des axes de rotation des roues.*

Tous les axes des roues ont pour point d'intersection le centre instantané de rotation. Ainsi, le vecteur vitesse en chaque point de la structure est orthogonal à la droite liant ce point au CIR. La figure 2.5 donne quelques exemples de centre instantané de rotation (lorsqu'il existe).

Proposition 2.3 *Pour qu'une disposition de roues soit viable et n'entraîne pas de glissement ou dérapage des roues sur le sol, il faut qu'il existe un unique CIR.*

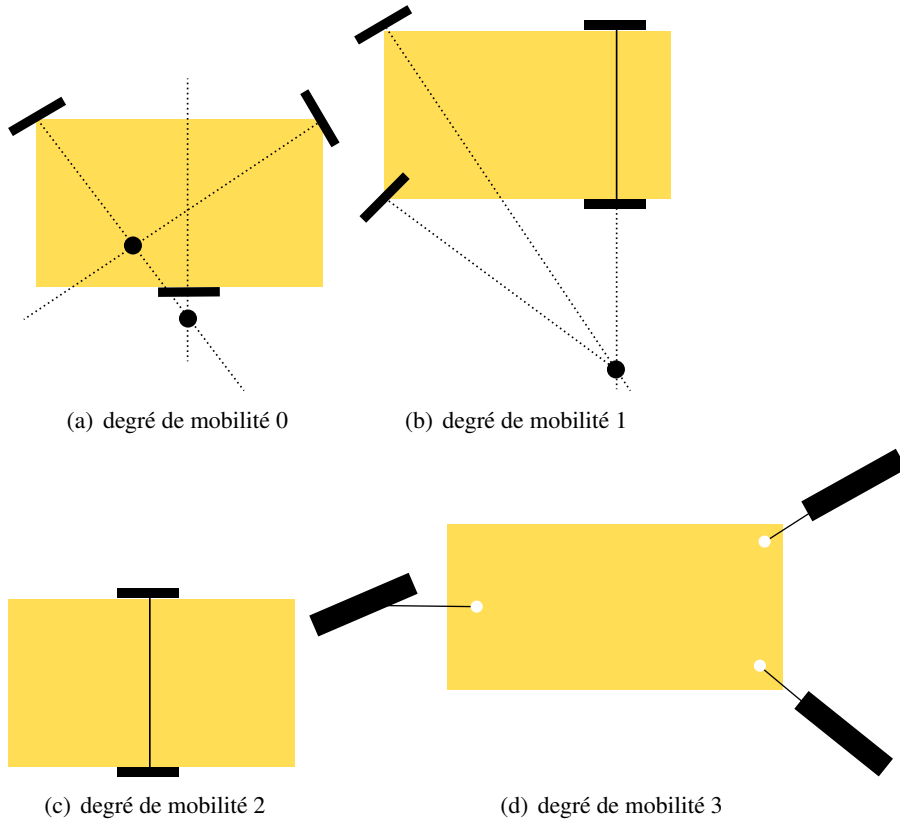


FIGURE 2.6 – Exemple pour chacun des degrés de mobilité possibles

Proposition 2.4 Pour que le CIR soit unique, il faut que $\text{rang}(C_1^*(\beta_{coc})) \leq 2$.

Pour cette raison, il existe en pratique 5 catégories de robots mobiles à roues (non dégénérés). Leur classification repose sur les notions de degré de mobilité et de degré de directionalité.

Définition 2.8 (Degré de mobilité) *Le degré de mobilité d'un robot est :*

$$\delta_m = \dim(\text{Ker}(C_1^*(\beta_{coc})) = 3 - \dim(\text{Im}(C_1^*(\beta_{coc})) = 3 - \text{rang}(C_1^*(\beta_{coc})).$$

Remarque 2.2 *Le degré de mobilité δ_m est le nombre de degré de liberté du mouvement du robot.*

Par exemple si le $\text{rang}(C_{1f})$ vaut deux, alors il y a au moins deux roues fixes et si en plus les axes de rotation des roues se croisent en un point CIR qui est fixe le seul mouvement possible est un mouvement circulaire (le robot tourne en rond). De même si le $\text{rang}(C_{1f})$ vaut un, alors les axes des roues fixes sont parallèles.

La figure 2.6 donne quatre exemples permettant d'illustrer chacun des cas allant d'un degré de mobilité nul à un degré de mobilité complet (3 dans le plan).

De la même façon, nous pouvons définir dans le cadre d'un robot non dégénéré le degré de directionalité. Dans ce cas nous avons

$$\text{rang}(C_{1f}) \leq 2 \quad (2.41)$$

$$\text{rang}(C_1^*(\beta_{coc}) = \text{rang}(C_{1f}) + \text{rang}(C_{1coc}(\beta_{coc})) \leq 2 \quad (2.42)$$

Définition 2.9 (Degré de directionalité) *Le degré de directionalité est défini par :*

$$\delta_s = \text{rang}(C_{1coc}(\beta_{coc}))$$

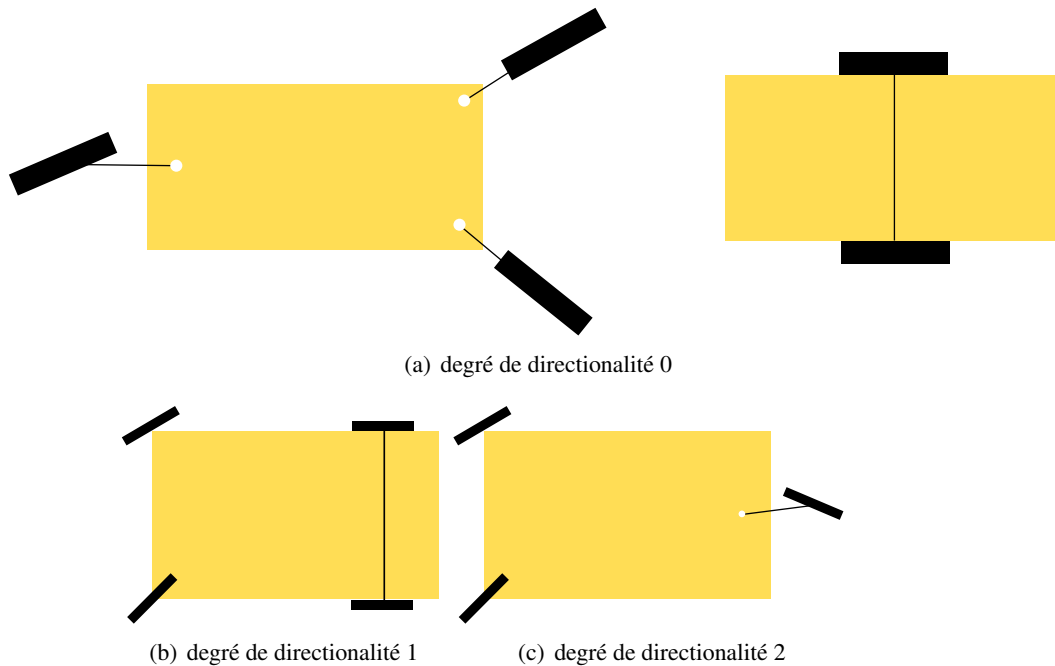


FIGURE 2.7 – Exemple pour chaque degré de directionalité

Notons que $\delta_s \leq 2$.

Le degré de directionalité δ_s est le nombre de roues centrées orientables pouvant être indépendantes pour diriger le robot.

2.1.4.6 Classification des robots mobiles à roues

Le degré de mobilité correspond au degré de liberté instantané du robot lié à la vitesse des roues tandis que le degré de directionalité correspond au degré de liberté instantané du robot lié à la possibilité d'orienter les roues (ce qui change le CIR). Le degré de manœuvrabilité correspond à la somme du degré de mobilité et du degré de directionalité.

Cinq structures non singulières existent dont le degré de mobilité δ_m et le degré de directionalité δ_s satisfont les trois inéquations suivantes :

$$1 \leq \delta_m \leq 3, \quad (2.43)$$

$$0 \leq \delta_s \leq 2, \quad (2.44)$$

$$2 \leq \delta_m + \delta_s \leq 3. \quad (2.45)$$

Suivant le critère de mobilité, deux classes importantes de robots mobiles apparaissent :

- Robots omnidirectionnels : mobilité totale dans le plan ($\delta_m = 3$),
- Robots à mobilité réduite : degré de mobilité inférieur à 3 ($\delta_m < 3$).

Les structures sont notées (δ_m, δ_s) , on parle alors de robot mobile de type (δ_m, δ_s) . Il existe donc seulement cinq types de robots mobiles non dégénérés, qui correspondent aux paires de valeurs (δ_m, δ_s) qui satisfont les trois inéquations (2.43). Ces combinaisons sont rappelées dans la table 2.2.

Type	I	II	III	IV	V
δ_m	3	2	2	1	1
δ_s	0	0	1	1	2

TABLE 2.2 – Types de robots mobiles [19]

Les caractéristiques principales de chaque type de robot mobile sont brièvement décrites par la suite (les exemples se concentrent sur des robots mobiles à trois roues).

Robot de type (3, 0). Ces robots n'ont ni roues conventionnelles fixes ($N_{cf} = 0$), ni roues conventionnelles orientables centrées ($N_{coc} = 0$).

Le robot est dit «omnidirectionnel» parce qu'il a une mobilité totale dans le plan, *i.e.* qu'il peut bouger dans n'importe quelle direction sans aucune réorientation du châssis à chaque instant. Inversement, les quatre autres types de robots mobiles ont une mobilité réduite. La figure 2.8 donne un exemple d'un robot de type (3, 0) composé de trois roues suédoises.

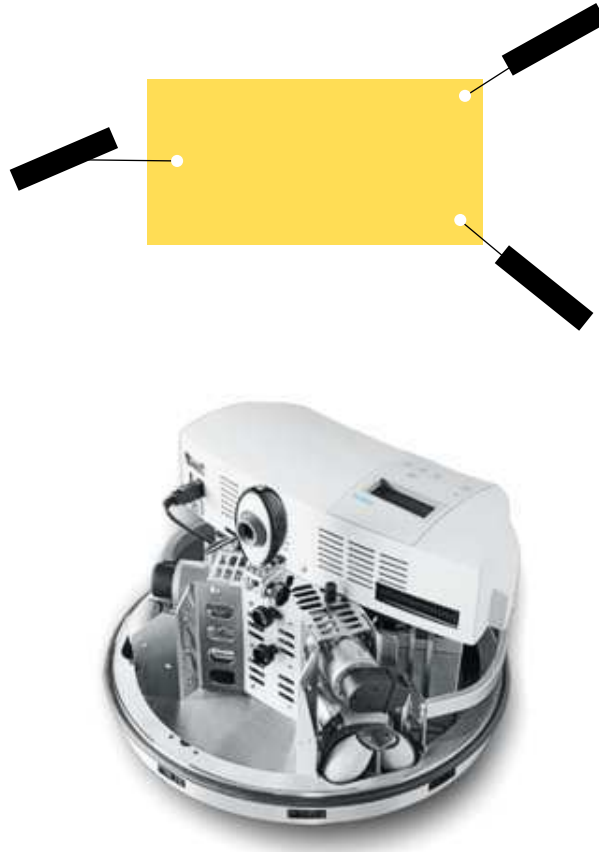


FIGURE 2.8 – Robot de type (3, 0)

Le modèle cinématique générique de posture pour ce type de robot est le suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \omega \end{pmatrix}, \quad (2.46)$$

Où v_1 est la vitesse selon l'axe \vec{x} , v_2 celle selon l'axe \vec{y} et ω la vitesse angulaire du robot.

Robot de type (2, 0). Ces robots n'ont pas de roues conventionnelles orientables centrées ($N_{coc} = 0$), mais ils ont une roue conventionnelle fixe ($N_{cf} > 0$), ou même plusieurs montées alors sur un seul axe commun. Le robot le plus connu en France appartenant à cette classe est le robot **Hilare** [57] dont une photo est donnée en figure 2.9. La figure 2.10 donne un exemple de robot de type (2, 0) composé de deux roues fixes de même axe et d'une roue orientable décentrée.

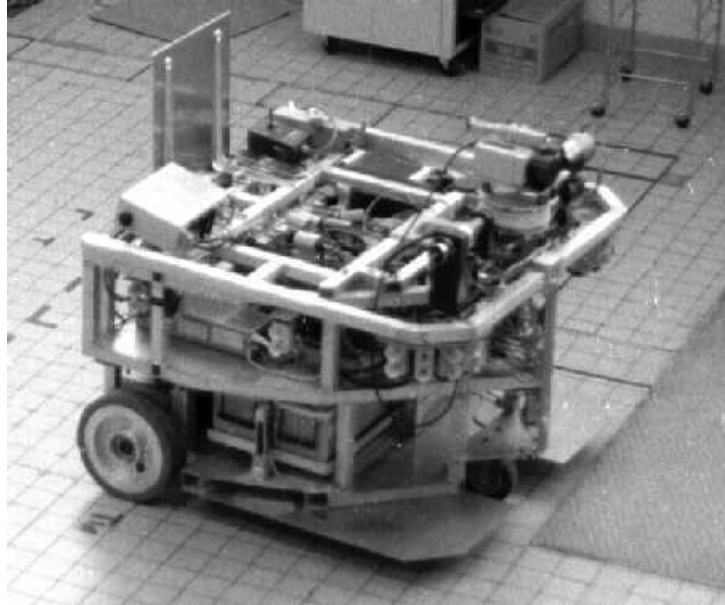


FIGURE 2.9 – Le robot HILARE du LAAS-CNRS



FIGURE 2.10 – Robot de type (2, 0)

Le modèle cinématique générique de posture pour ce type de robot est le suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}, \quad (2.47)$$

Où v est la vitesse linéaire du robot et ω sa vitesse angulaire.

Robot de type (2, 1). Ces robots n'ont pas de roues conventionnelles fixes ($N_{cf} = 0$). Ils ont une roue conventionnelle orientable centrée ($N_{coc} = 1$) et deux roues conventionnelles orientables décentrées ($N_{cod} = 2$). La figure 2.11 donne un exemple d'un robot de type (2, 1).

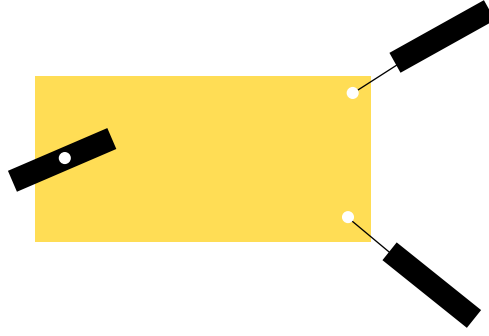


FIGURE 2.11 – Robot de type (2, 1)

Le modèle cinématique générique de posture pour ce type de robot est le suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta}_{c1} \end{pmatrix} = \begin{pmatrix} \cos(\theta + \beta_{c1}) & 0 & 0 \\ \sin(\theta + \beta_{c1}) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega_1 \\ \omega_2 \end{pmatrix}, \quad (2.48)$$

Où v est la vitesse linéaire du robot, ω_1 sa vitesse angulaire et ω_2 la vitesse angulaire de sa roue orientable centrée.

Robot de type (1, 1). Ces robots ont une ou plusieurs roues conventionnelles fixes ($N_{cf} \geq 1$) avec un seul axe commun et une ou plusieurs roues conventionnelles orientables centrées ($N_{coc} \geq 1$). La figure 2.12 donne deux exemples de robots de type (1, 1).

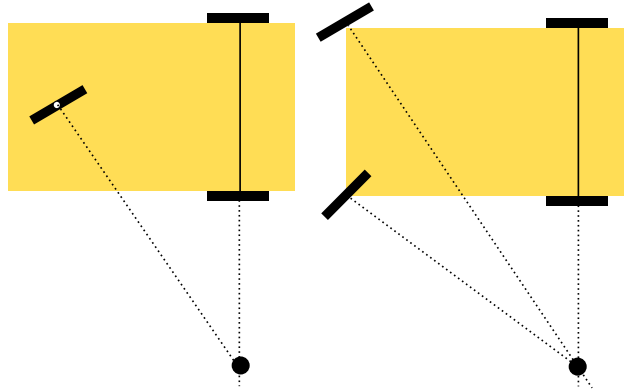


FIGURE 2.12 – Robot de type (1, 1)

Le modèle cinématique générique de posture pour ce type de robot est le suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta}_{c3} \end{pmatrix} = \begin{pmatrix} L \cos \theta \sin \beta_{c3} & 0 \\ L \sin \theta \sin \beta_{c3} & 0 \\ \cos \beta_{c3} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}, \quad (2.49)$$

Où v est équivalent à une vitesse linéaire et ω à la vitesse angulaire du robot.

Robot de type (1, 2). Ces robots n'ont pas de roues conventionnelles fixes ($N_{cf} = 0$). Ils ont au minimum deux roues conventionnelles orientables centrées

($N_{coc} \geq 2$). La figure 2.13 donne un exemple d'un robot de type (1, 2) avec deux roues conventionnelles orientables centrées et une roue décentrée orientable.



FIGURE 2.13 – Robot de type (1, 2)

Le modèle cinématiques générique de posture pour ce type de robot est le suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta}_{c1} \\ \dot{\beta}_{c2} \end{pmatrix} = \begin{pmatrix} (\sin \beta_{c1} \cos(\theta \beta_{c2}) + \sin \beta_{c2} \cos(\theta + \beta_{c1})) & 0 & 0 \\ (\sin \beta_{c1} \sin(\theta \beta_{c2}) + \sin \beta_{c2} \sin(\theta + \beta_{c1})) & 0 & 0 \\ \sin(\beta_{c2} - \beta_{c1}) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega_1 \\ \omega_2 \end{pmatrix}, \quad (2.50)$$

Où v est équivalent à une vitesse angulaire. ω_1 et ω_2 sont les vitesses angulaires des deux roues orientables centrées.

2.2 OBSERVABILITÉ, PLATITUDE ET ALGÈBRE DIFFÉRENTIELLE

L'algèbre différentielle a été introduite en automatique au milieu des années quatre-vingt à travers les travaux de M. Fliess [42], T. Glad et L. Ljung [89]. Des développements et des évolutions du point de vue algorithmique par exemple ont été introduits par S. Diop [36], F. Ollivier [116] et K. Forsman [53]. Ces différentes évolutions ont provoqué un regain d'intérêt pour ces méthodes et ont amené de nombreux nouveaux développements [125, 9, 131, 67, 118, 55, 128, 113, 95, 104, 133]. Nous allons ici rappeler les principaux résultats d'algèbre différentielle nécessaires à la bonne compréhension de ce travail, notamment les résultats sur la platitude des systèmes non-linéaires. L'algèbre différentielle prend corps sur la théorie de l'algèbre dont les principaux concepts et définitions sont rappelés dans l'annexe A.1. Les différentes définitions proviennent du cours de Rosanne Ushirobira et Wilfrid Perruquetti.

Définition 2.10 (Anneaux (ou corps) différentiel) *Un anneau différentiel (ou corps différentiel) \mathbf{K} est un anneau (ou corps) muni d'une dérivation, notée $\frac{d}{dt}$, c'est-à-dire une application $\frac{d}{dt} : \mathbf{K} \rightarrow \mathbf{K}$, qui vérifie :*

$$\forall a, b \in \mathbf{K}, \frac{d}{dt}(a + b) = \frac{d}{dt}(a) + \frac{d}{dt}(b) \quad (2.51)$$

$$\forall a, b \in \mathbf{K}, \frac{d}{dt}(ab) = \frac{d}{dt}(a)b + a\frac{d}{dt}(b) \quad (2.52)$$

Exemple 2.5 On peut définir une dérivation sur $\mathbb{C}[X]$ par :

$$\partial_{\mathbb{C}[x]} : \mathbb{C}[x] \rightarrow \mathbb{C}[x] \quad (2.53)$$

$$\sum_{j=0}^n a_j x^j \rightarrow \sum_{j=1}^n j a_j x^{j-1} \quad (2.54)$$

Définition 2.11 (Extension de corps différentiel) Une **extension de corps différentiel** \mathbf{L}/\mathbf{K} est la donnée de deux corps différentiels \mathbf{K} et \mathbf{L} tels que \mathbf{K} est un sous corps de \mathbf{L} et la restriction de la dérivation de \mathbf{L} à \mathbf{K} coïncide avec la dérivation de \mathbf{K} .

Exemple 2.6 La dérivation de l'exemple 2.5 peut être étendue sur $\mathbb{C}(X)$ le corps des fractions des polynômes de \mathbb{C} .

Soit $h \in \mathbb{C}(X)$, on a donc $f, g \in \mathbb{C}[X]$ tels que $h = \frac{f}{g}$ où $g \neq 0$, on définit alors :

$$\partial_{\mathbb{C}(X)} : \mathbb{C}(X) \rightarrow \mathbb{C}(X) \quad (2.55)$$

$$h = \left(\frac{f}{g} \right) \rightarrow \frac{\partial_{\mathbb{C}[X]}(f)g - f\partial_{\mathbb{C}[X]}(g)}{g^2} \quad (2.56)$$

Définition 2.12 Soit \mathbf{S} un sous-ensemble de \mathbf{L} . Le corps engendré par \mathbf{S} sur \mathbf{K} est le plus petit (par rapport à l'inclusion) sous corps de \mathbf{L} contenant à la fois \mathbf{K} et \mathbf{S} . On le note $\mathbf{K}(\mathbf{S})$. L'extension de corps différentiel \mathbf{L}/\mathbf{K} est dite simple, si \mathbf{S} ne contient qu'un seul élément, elle est dite finiment engendrée (ou de type fini) si \mathbf{S} contient un nombre fini d'éléments.

Remarque 2.3 Prenons \mathbf{L}/\mathbf{K} une extension différentiel. Prenons $\alpha \in \mathbf{L}$. Le sous-ensemble différentiel généré par α sur \mathbf{L} est noté $\mathbf{L} \langle \alpha \rangle$. Il comprend tous les éléments de la forme $P(\alpha)$ où P est un polynôme différentiel dans $\mathbf{K}[X]$. De même, pour $\mathbf{S} = \{\alpha_1, \dots, \alpha_\ell\} \subset \mathbf{L}$, le sous-ensemble différentiel généré par \mathbf{S} sur \mathbf{L} est noté $\mathbf{L} \langle \mathbf{S} \rangle$. Il comprend tous les éléments de la forme $P(\alpha_1, \dots, \alpha_\ell)$ où P est un polynôme différentiel dans $\mathbf{L}[X_1, \dots, X_\ell]$.

Exemple 2.7 L'extension de corps \mathbb{C}/\mathbb{R} est finiment engendrée, car $\mathbb{C} = \mathbb{R}(\sqrt{-1})$.

Définition 2.13 (Clôture algébrique) Soient \mathbf{K} un corps et \mathbf{L}/\mathbf{K} une extension de corps algébrique. Alors si tout polynôme $P \in \mathbf{K}[X]$ s'écrit :

$$P = a_0(X - \alpha_1) \dots (X - \alpha_n), a_0 \in \mathbf{K}, \alpha_i \in \mathbf{L} \quad (2.57)$$

on appelle \mathbf{L} une clôture algébrique de \mathbf{K} . Si on peut choisir $\mathbf{L} = \mathbf{K}$, le corps \mathbf{K} est dit algébriquement clos.

Chaque corps admet une clôture algébrique, qui est déterminée à un isomorphisme sur \mathbf{K} près. On écrit $\overline{\mathbf{K}}$ pour cette clôture algébrique unique (à un isomorphisme près).

Remarque 2.4 Un corps \mathbf{K} est dit algébriquement clos si tout $P(X) \in \mathbf{K}[X]$ non constant a au moins une racine dans \mathbf{K} .

Exemple 2.8 Le théorème de d'Alembert-Gauss permet de dire qu'une clôture algébrique du corps des réels est le corps des complexes.

Définition 2.14 (Différentiellement algébrique / Différentiellement transcendant) Soit \mathbf{L}/\mathbf{K} une extension différentielle. Un élément a de \mathbf{L} est dit **différentiellement algébrique** sur \mathbf{K}

s'il satisfait une équation différentielle algébrique à coefficients dans \mathbf{K} :

$$\exists P(\mathbf{X}, \dots, \mathbf{X}^{(\alpha)}) \neq 0 \text{ tel que } P(a, \dots, a^{(\alpha)}) = 0 \quad (2.58)$$

où P est un polynôme différentiel irréductible (cela signifie qu'il ne peut pas être écrit comme un produit de deux polynômes non triviaux). Si $a \in \mathbf{L}$ ne satisfait aucune équation différentielle algébrique à coefficients dans \mathbf{K} , a est **différentiellement transcendant** sur \mathbf{K} . Si tout élément de \mathbf{L} est algébrique sur \mathbf{K} , l'extension de corps \mathbf{L}/\mathbf{K} est dite **différentiellement algébrique** sinon elle est dite **différentiellement transcendante**.

- Exemple 2.9**
- Algébrique : La racine $a = \sqrt{2} \in \mathbb{R}$ est algébrique sur \mathbb{Q} , car $P(a) = 0$ pour $P(Z) = Z^2 - 2$,
 - Transcendant : On sait que π est transcendant sur \mathbb{Q} (la démonstration originale est de Carl Louis Ferdinand von Lindemann en 1882 et a permis de prouver l'impossibilité du problème de la quadrature du cercle (qui consiste à construire un carré de même aire qu'un cercle donné à l'aide d'une règle et d'un compas),
 - Différentiellement algébrique : On voit que $a = \sin(t) \in \mathbb{Q} \langle \sin(t) \rangle$ est différentiellement algébrique sur \mathbb{Q} car $P(a, \dot{a}) = 0$ pour $P(Z, \dot{Z}) = Z^2 + \dot{Z}^2 - 1$.

Définition 2.15 Une famille $\mathbf{S} = \{\alpha_1, \dots, \alpha_\ell\} \subset \mathbf{L}$ est dite algébriquement indépendante sur \mathbf{K} s'il n'existe aucun polynôme $P(\mathbf{S}) \in \mathbf{K}[\mathbf{S}]$, $P \neq 0$, tel que $P(\mathbf{S}) = 0$. Dans le cas contraire on dit que \mathbf{S} est \mathbf{K} -algébriquement dépendant.

Définition 2.16 (Base de transcendance différentielle / Degré de transcendance différentielle) Une base de l'extension différentielle \mathbf{L}/\mathbf{K} est une famille algébriquement différentiellement indépendante sur \mathbf{K} , maximale par rapport à l'inclusion. Le nombre d'éléments d'une telle base est appelé le degré de transcendance différentielle et est noté $d^\circ \text{ tr diff } \mathbf{L}/\mathbf{K}$.

Remarque 2.5 Prenons $\mathbf{F}/\mathbf{K}/\mathbf{F}$ une tour d'extension différentielle. Nous avons :

$$d^\circ \text{ tr diff } \mathbf{L}/\mathbf{F} = d^\circ \text{ tr diff } \mathbf{L}/\mathbf{K} + d^\circ \text{ tr diff } \mathbf{K}/\mathbf{F} \quad (2.59)$$

Proposition 2.5 Soit \mathbf{L}/\mathbf{K} une extension différentielle finiment engendrée. Les deux propriétés suivantes sont équivalentes :

1. l'extension \mathbf{L}/\mathbf{K} est différentiellement algébrique : $d^\circ \text{ tr diff } \mathbf{L}/\mathbf{K} = 0$,
2. le degré de transcendance (non différentiel) de l'extension \mathbf{L}/\mathbf{K} est fini.

2.2.1 Système et dynamiques

Considérons un système physique représenté sous sa forme mathématique par r équations différentielles ordinaires algébriques. Ce système mathématique lie un nombre fini de variables w_1, \dots, w_n avec $r < n$. Il peut être écrit sous la forme suivante :

$$P_\ell(w_1, \dots, w_1^{\mu_{(1,1)}}, \dots, w_n, \dots, w_n^{\mu_{(1,n)}}) = 0, \ell = 1, \dots, r \quad (2.60)$$

Où $P_\ell, \ell = 1, \dots, r$ sont des polynômes et $\mu_{(l,i)} \in \mathbb{N}, i = 1, \dots, n$

Remarque 2.6 *Il est important de noter que tout système physique rencontré dans le monde réel peut être représenté sous cette forme, les autres termes non linéaires pouvant être réécrits sous cette forme.*

Exemple 2.10 *Afin d'illustrer cette partie, reprenons le modèle d'état d'un robot unicycle (2.47). Ce modèle peut être réécrit sous forme algébrique :*

$$\boxed{\begin{cases} \dot{x} = v \cos(\theta) \\ \dot{y} = v \sin(\theta) \\ \dot{\theta} = \omega \end{cases}} \Rightarrow \begin{cases} z = x + iy \\ \Theta = \exp(i\theta) \end{cases} \Rightarrow \boxed{\begin{cases} \dot{z} = \Theta v \\ \dot{\Theta} = i\Theta\omega \end{cases}} \quad (2.61)$$

2.2.1.1 Système non linéaire

Un système non linéaire est une extension de corps \mathbf{L}/\mathbf{K} finiment engendré.

- \mathbf{K} est appelé **corps de base** associé au système. Il contient les coefficients de n'importe quel système d'équations correspondant au système \mathbf{L}/\mathbf{K} ,
- Le corps \mathbf{L} est appelé **corps du système** ou corps systémique, il contient les variables et toutes les expressions algébriques qui peuvent être construites à partir des variables du système.

De manière sous-jacente à l'extension, il est possible de construire un système d'équations algébro-différentielles polynomiales, qui sont les équations satisfaites par les variables du système.

Exemple 2.11 *L'équation finale de (2.61) étant complexe, prenons $\mathbf{K} = \mathbb{C}$.*

Pour chacune des variables du système, (z, Θ, v, ω) associons-lui une variable indéterminée, $z \rightarrow \mathbf{X}_1, \Theta \rightarrow \mathbf{X}_2, v \rightarrow U_1, \omega \rightarrow U_2$.

Nous pouvons générer le corps différentiel suivant : $\mathbb{C} \langle \mathbf{X}_1, \mathbf{X}_2, U_1, U_2 \rangle$ qui est le corps différentiel engendré par \mathbb{C} et $\{\mathbf{X}_1, \mathbf{X}_2, U_1, U_2\}$. On pose :

$$P_1 = \dot{\mathbf{X}}_1 - \mathbf{X}_2 U_1, \quad (2.62)$$

$$P_2 = \dot{\mathbf{X}}_2 - i\mathbf{X}_2 U_2. \quad (2.63)$$

Construisons l'idéal défini à partir de P_1 et P_2 et notons le $I\{P_1, P_2\}$. D'après[27], cet idéal est premier. D'après la proposition A.1, l'anneau quotient $\mathbb{C}\{\mathbf{X}_1, \mathbf{X}_2, U_1, U_2\}/I\{P_1, P_2\}$ est un domaine intègre. Nous pouvons donc considérer le corps des fractions différentielles associé noté \mathbf{L} (c'est l'ensemble des fractions de polynômes de $\mathbb{C} \langle \mathbf{X}_1, \mathbf{X}_2, U_1, U_2 \rangle$ dont le dénominateur n'appartient pas à I et est non nul, par exemple, $X_1 = \frac{X_1 + I}{1 + I} \in \mathbf{L}$.

2.2.1.2 Entrées du système

Définition 2.17 (Entrée) *Une **entrée** du système \mathbf{L}/\mathbf{K} est un ensemble fini $\mathbf{e} = (\mathbf{e}_1, \dots, \mathbf{e}_s)$ de \mathbf{L} tel que l'extension $\mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle$ est différentiellement algébrique.*

Une entrée est dite indépendante, si c'est un ensemble différentiellement algébrique indépendant sur \mathbf{K} . Dans ce cas :

$$\begin{aligned} d^o \text{ tr diff } \mathbf{L}/\mathbf{K} &= d^o \text{ tr diff } \mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle + d^o \text{ tr diff } \mathbf{K} \langle \mathbf{e} \rangle / \mathbf{K} \\ &= d^o \text{ tr diff } \mathbf{K} \langle \mathbf{e} \rangle / \mathbf{K}. \end{aligned}$$

En effet $d^o \text{ tr diff } \mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle = 0$, car par définition il est différentiellement algébrique. \mathbf{e} est donc une base de l'extension différentielle \mathbf{L}/\mathbf{K} .

Notons que $d^o \text{tr diff } \mathbf{L}/\mathbf{K}$ (donc le nombre d'entrées) est la différence entre le nombre de variables indépendantes et le nombre d'équations différentielles indépendantes.

Exemple 2.12 *Le robot unicycle a 4 variables indépendantes et 2 équations différentielles indépendantes donc $d^o \text{tr diff } \mathbf{L}/\mathbf{K} = 2$. Le robot possède donc deux entrées indépendantes, v et ω*

2.2.1.3 Dynamique du système

Définition 2.18 (Dynamique) *Une **dynamique** est un système \mathbf{L}/\mathbf{K} dont on a distingué, en général pour des raisons liées au fonctionnement du système, une entrée particulière. On note une dynamique sous la forme $\mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle$. Cela signifie que l'on considère le système \mathbf{L}/\mathbf{K} ayant pour entrée $\mathbf{e} = (\mathbf{e}_1, \dots, \mathbf{e}_s)$.*

2.2.1.4 Sorties du système

Définition 2.19 (Sortie) *Une **sortie** d'un système \mathbf{L}/\mathbf{K} est un ensemble fini $\mathbf{y} = (y_1, \dots, y_p)$ de \mathbf{L} .*

N'importe quelle famille finie de \mathbf{L} peut être considérée comme une sortie. Ce sont des raisons liées au fonctionnement du système (par exemple des contraintes technologiques liées aux capteurs) ou à son contrôle qui déterminent le choix de la sortie. On distingue en général les sorties de commandes, qui correspondent aux variables que l'on cherche à piloter, et les sorties de mesures qui correspondent aux variables mesurées par les capteurs.

Exemple 2.13 *Pour le robot unicycle nous pouvons prendre par exemple $\mathbf{y} = z$ comme sortie de notre système. Nous verrons d'ailleurs que cette sortie joue un rôle particulier pour le système (c'est une sortie plate). Il faut noter que cette sortie nécessite deux capteurs pour récupérer la position $z = x + iy$.*

2.2.1.5 État et réalisation

Comme l'extension \mathbf{L}/\mathbf{K} est finiment engendrée et comme pour tout choix d'entrée \mathbf{e} de \mathbf{L}/\mathbf{K} , l'extension $\mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle$ est différentiellement algébrique, l'extension différentielle $\mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle$ possède, d'après la proposition 2.5, un degré de transcendance fini. Cela permet de définir la notion d'état.

Définition 2.20 (Etat) *Un **état** de la dynamique $\mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle$ est une base de transcendance $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ de l'extension $\mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle$.*

Le nombre n d'états nécessaires à la description d'un système est appelé la dimension, avec $n = d^o \text{tr } \mathbf{L}/\mathbf{K} \langle \mathbf{e} \rangle$ (degré de transcendance non différentielle). Ainsi en fonction de l'entrée choisie, la dimension de l'état peut être différente.

2.2.2 Étude des propriétés d'un système non-linéaire

Dans cette section, la notion de platitude différentielle sera introduite. Cette notion est liée à la notion d'observabilité des systèmes. Ainsi il sera possible d'étudier l'observabilité (et donc la localisabilité) d'un système en utilisant les notions d'algèbres différentielles introduites dans ce chapitre.

2.2.2.1 Platitude différentielle

Définition 2.21 (Platitude différentielle) *Un système \mathbf{L}/\mathbf{K} est dit **différentiellement plat** si à une clôture algébrique (non différentielle) près, \mathbf{L} est un corps d'extension purement transcendant de \mathbf{K} . Autrement dit, \mathbf{L}/\mathbf{K} est différentiellement plat s'il existe une base de transcendance différentielle $y = (y_1, \dots, y_m)$ de $\overline{\mathbf{L}}/\mathbf{K}$, pour laquelle $\overline{\mathbf{L}} = \overline{\mathbf{K}} \langle y \rangle$.*

Le nombre des composantes d'une sortie plate est égal au degré de transcendance différentielle de l'extension \mathbf{L}/\mathbf{K} . Si le système admet p entrées indépendantes, alors nous avons vu que ces entrées forment une base de transcendance différentielle de \mathbf{L}/\mathbf{K} . Ainsi, il y a autant de sorties plates que d'entrées indépendantes au système.

Nous pouvons à partir de la définition donner trois critères pour expliciter la platitude :

- La sortie plate est fonction des variables du système,
- Il n'existe aucune relation algébrique reliant les sorties plates entre elles,
- Toute variable du système s'exprime en fonction des sorties plates et de leurs dérivées.

Exemple 2.14 *Le robot unicycle ayant deux entrées indépendantes (sur \mathbb{R}), il est nécessaire de choisir deux sorties plates (sur \mathbb{R}). Soit le robot unicycle décrit par le modèle (2.47). Prenons $z = x + iy$ comme sortie plate (qui correspond à deux sorties plates dans \mathbb{R}). Il est clair qu'il n'existe pas de relation algébrique entre x et y et que toute variable du système peut s'exprimer en fonction de z et de ses dérivées :*

$$\begin{cases} z & = & z \\ \Theta & = & \frac{\dot{z}}{|\dot{z}|} \\ v & = & |\dot{z}| \\ \omega & = & -i \frac{\ddot{z}|\dot{z}|^2 - \dot{z}^2 \bar{\ddot{z}}}{2|\dot{z}|^2 \dot{z}} \end{cases} \quad (2.64)$$

Le robot unicycle est donc plat et sa sortie plate est z .

Ces relations découlent des calculs suivant :

On a $\dot{z} = \Theta v$ et $|\Theta| = 1$ donc $|v| = |\dot{z}| \in \mathbb{R}$. Ainsi, $\Theta = \pm \frac{\dot{z}}{|\dot{z}|}$. On a $v = \pm |\dot{z}|$.

Par ailleurs, on sait que (où \bar{u} est le complexe conjugué de u) :

$$\frac{d|u|}{dt} = \frac{\dot{u}\bar{u} + u\dot{\bar{u}}}{2|u|}$$

que l'on utilise pour calculer $\frac{d|\dot{z}|}{dt}$ ce qui permet de calculer $\omega = -i \frac{\dot{\Theta}}{\Theta}$:

$$\omega = -i \times \frac{\ddot{z}|\dot{z}| - \dot{z} \left[\frac{\ddot{z}\bar{\dot{z}} + \dot{z}\bar{\ddot{z}}}{2|\dot{z}|} \right]}{|\dot{z}|^2} \times \frac{|\dot{z}|}{\dot{z}} \quad (2.65)$$

$$= -i \times \frac{2\ddot{z}|\dot{z}|^2 - |\dot{z}|^2 \ddot{z} - \dot{z}^2 \bar{\ddot{z}}}{2|\dot{z}|^2 \dot{z}} \quad (2.66)$$

$$= -i \times \frac{\ddot{z}|\dot{z}|^2 - \dot{z}^2 \bar{\ddot{z}}}{2|\dot{z}|^2 \dot{z}} \quad (2.67)$$

2.2.2.2 Inversion

Soit L/K un système avec $d^0 \text{ tr diff } L/K = m$ et soit $y = (y_1, \dots, y_p)$ une sortie.

Définition 2.22 (Rang différentiel de sortie) *Le degré de transcendance différentielle de $K \langle y \rangle / K$ est appelé le **rang différentiel de sortie** du système L/K (par rapport à $K \langle y \rangle / K$, ou à y), que l'on note $\rho_y = d^0 \text{ tr diff } K \langle y \rangle / K$.*

Définition 2.23 (Inversibilité) *Le système L/K avec la sortie y est dit **inversible à droite** si le rang différentiel de sortie est égal au nombre de composantes de la sortie : $\rho_y = p$. Il est dit **inversible à gauche** si le rang de sortie est égal au nombre de composantes indépendantes de l'entrée : $\rho_y = m = d^0 \text{ tr diff } L/K$. Le système est dit **inversible** s'il est inversible à droite et à gauche.*

Exemple 2.15 *Pour le robot unicycle avec la sortie plate $y = z = x + iy$, il a été montré que toute équation de K pouvait être exprimée en fonction de y , le rang différentiel de sortie est donc égal à 2 ainsi que le nombre de sorties. Le système est donc inversible à droite. Le système a deux entrées donc il est aussi inversible à gauche. Le système est donc inversible.*

Nous aurions pu prévoir ce résultat ; en effet, nous allons voir que tout système plat est inversible par rapport à toute sortie plate.

2.2.2.3 Liens entre inversion et platitude

Proposition 2.6 *Un système plat est inversible par rapport à toute sortie plate y .*

Définition 2.24 (Défaut de platitude) *Soit $\Gamma \subseteq \bar{L}$ un sur-corps différentiel de K , tel qu'à une clôture algébrique près, l'extension Γ/K soit différentiellement purement transcendante et le degré de transcendance (non différentiel), $d^0 \text{ tr } \bar{L}/\Gamma$ soit minimal (entre ces extensions). Alors on appelle $d^0 \text{ tr } \bar{L}/\Gamma$ le **défaut** du système.*

Remarque 2.7 *Si z est une base de transcendance différentielle de Γ/K , le défaut est égal à la dimension d'état de la dynamique inverse par rapport à z . Pour une paramétrisation complète du système, on doit spécifier, en plus des m trajectoires pour une sortie plate, au maximum $d^0 \text{ tr } \bar{L}/\Gamma$ trajectoire(s) supplémentaire(s). Toutefois, une telle paramétrisation n'est pas libre.*

2.2.2.4 Observabilité

Les différentes définitions proviennent du cours de Rosanne Ushirobira et Wilfrid Perruquetti.

Il n'est pas possible, en général, d'utiliser autant de capteurs que de variables d'état nécessaires pour caractériser le comportement du système, pour des raisons par exemple de coût, ou encore à cause de contraintes technologiques. La taille du vecteur de sortie de mesure est donc en général plus petite que celle du vecteur d'état. Dans ce cas, à un instant t donné, l'état $x(t)$ ne peut pas être algébriquement déduit des variables de sortie y (observés à l'instant t).

Cependant la connaissance de la variation temporelle de l'état est indispensable dans des domaines tels que la modélisation (pour l'identification des paramètres), le «monitoring» (détection de fautes) ou le contrôle des systèmes. C'est pourquoi la reconstruction de l'état en utilisant un observateur est fondamentale.

Définition 2.25 (Observateur/ Observateur à entrées inconnues) *L'objectif d'un **observateur d'état** est d'estimer à chaque instant l'état $\mathbf{x}(t)$ à partir de la sortie mesurée $\mathbf{y}(t)$ et de l'entrée $\mathbf{u}(t)$. Dans un cadre plus général un **observateur d'état à entrées inconnues** doit permettre d'estimer à chaque instant l'état $\mathbf{x}(t)$ à partir de la sortie mesurée $\mathbf{y}(t)$ et sans connaissance de l'entrée $\mathbf{u}(t)$ (dans certains cas il est possible d'estimer l'entrée $\mathbf{u}(t)$ à partir de la sortie $\mathbf{y}(t)$). Cette observation n'est possible que si le système est observable.*

Le cadre algébrique présenté dans cette section se prête parfaitement à l'étude de l'observabilité d'un système en linéaire/non linéaire.

Théorème 2.2 (Observabilité) *Prenons \mathbf{L}/\mathbf{K} un système non linéaire et $\mathbf{z} = \{z_\ell, \ell = 1, \dots, p\}$ un sous-ensemble de \mathbf{L} . Un élément ξ de \mathbf{L} est dit observable par rapport à \mathbf{z} s'il est différentiellement algébrique sur $\mathbf{K} \langle \mathbf{z} \rangle$. Un sous-ensemble $\mathbf{S} \subset \mathbf{L}$ est dit observable par rapport à \mathbf{z} si et seulement si tous ses éléments sont observables par rapport à \mathbf{z} .*

Un élément ξ est observable par rapport à \mathbf{z} s'il peut être obtenu comme une fonction algébrique des composantes de \mathbf{z} et d'un nombre fini de leurs dérivées. Cette notion d'observabilité englobe aussi la notion d'identifiabilité. En général, pour étudier l'observabilité, il est possible de prendre $\mathbf{z} = \{\mathbf{u}, \mathbf{y}\}$ avec \mathbf{u} l'entrée et \mathbf{y} la sortie (des vecteurs). Dans ce cas il est possible de prendre \mathbf{S} comme l'ensemble de toutes les variables d'état. Dans ce cas de figure, le système est observable si, et seulement si, toutes les variables d'état peuvent être exprimées comme une fonction algébrique des variables d'entrées et de sorties du système et de leur dérivées jusqu'à un ordre fini.

Exemple 2.16

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(\dot{x}, \dot{u}) \\ y = x_1 \end{cases} \quad (2.68)$$

Ce système est toujours observable et ce quelque soit la fonction f . En effet, $x_1 = y$ et $x_2 = \dot{y}$. En fait, x_1 est la sortie plate du système et comme le montrera la proposition 2.8, un système est toujours observable par sa sortie plate.

Proposition 2.7 (Observabilité par une sortie plate) *Un système plat \mathbf{L}/\mathbf{K} est observable par n'importe quelle sortie plate.*

Exemple 2.17 *Reprenons l'exemple du robot unicycle. Supposons que la sortie mesurée soit $z = x + iy$ (notons qu'en général, x et y sont mesurées indépendamment l'un de l'autre). Il est évident par les équations de platitude (voir exemple 2.14) que le robot est observable.*

Nous pouvons noter que dans l'exemple précédent, non seulement la connaissance de l'entrée v, ω n'est pas nécessaire pour estimer l'état mais plus encore, ces entrées peuvent être observées en même temps que l'état. Nous avons donc un estimateur d'état et d'entrée.

Proposition 2.8 *L'état et les entrées d'un système plat \mathbf{L}/\mathbf{K} sont observables par n'importe quelle sortie plate.*

Proposition 2.9 *Soit \mathbf{L}/\mathbf{K} et \mathbf{z} une sortie. Le défaut de platitude du système par rapport à \mathbf{z} (voir la*

définition 2.24) donne le nombre d'équations algébriques indépendantes à ajouter pour obtenir l'observabilité à entrée inconnue du système.

2.2.3 Platitude dans le cas des robots mobiles à roues

Remarque 2.8 *L'étude à venir est faite dans le cas où la vitesse du robot est positive, il est facile de généraliser l'étude pour une vitesse négative en changeant le signe lorsque nécessaire (le signe de la vitesse peut être facilement déterminé).*

Avant une étude de chacune des classes notons que les cinq modèles représentatifs de chacune des classes sont plats. Ceci découle d'un résultat fondamental du à P. Martin et P. Rouchon [98] (voir aussi [96, 97])

Théorème 2.3 *Tout système commandable sans dérive ayant m entrées et au plus $m + 2$ variables d'état est plat c'est-à-dire de la forme : $\dot{x} = B(x)u, x \in \mathbb{R}^n, u \in \mathbb{R}^m$ avec $n \leq m + 2$*

En effet on a :

Type	I	II	III	IV	V
m	3	2	3	2	3
n	3	3	4	4	5

TABLE 2.3 – Nombre d'entrées et d'états pour les cinq types de robots mobiles

2.2.3.1 Robot de type (2, 0)

Modèle cinématique : Le modèle cinématique est le suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}, \quad (2.69)$$

où v est la vitesse linéaire du robot et ω sa vitesse angulaire. Nous avons déjà vu qu'en posant

$$\begin{cases} z &= x + iy \\ \Theta &= \exp(i\theta) \end{cases}, \quad (2.70)$$

le modèle cinématique d'un robot de type (2, 0) peut être réécrit sous :

$$\begin{pmatrix} \dot{z} \\ \dot{\Theta} \end{pmatrix} = \begin{pmatrix} \Theta & 0 \\ 0 & i\Theta \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (2.71)$$

Platitude : Un robot de type (2, 0) admet z comme sortie plate et les équations de platitude sont alors données par :

$$z = z, \quad (2.72)$$

$$\Theta = \pm \frac{\dot{z}}{|\dot{z}|}, \quad (2.73)$$

$$v = \pm |\dot{z}|, \quad (2.74)$$

$$\omega = \mp i \times \frac{\ddot{z}|\dot{z}|^2 - \dot{z}^2\bar{\ddot{z}}}{2|\dot{z}|^2\dot{z}}, \quad (2.75)$$

les indéterminations sur les signes se lèvent facilement avec de simples considérations de bon sens physique.

2.2.3.2 Robot de type (3, 0)

Modèle cinématique : Le modèle cinématique est le suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \omega \end{pmatrix}, \quad (2.76)$$

où v_1 est la vitesse selon l'axe \vec{x} , v_2 celle selon l'axe \vec{y} et ω la vitesse angulaire du robot. En posant

$$\begin{cases} z = x + iy \\ \Theta = \exp(i\theta) \end{cases}, \quad (2.77)$$

le modèle cinématique (2.76) peut être alors être réécrit sous la forme suivante :

$$\begin{pmatrix} \dot{z} \\ \dot{\Theta} \end{pmatrix} = \begin{pmatrix} \Theta & i\Theta & 0 \\ 0 & 0 & i\Theta \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \omega \end{pmatrix}. \quad (2.78)$$

En prenant comme nouvelle entrée $v = v_1 + iv_2$, le modèle devient semblable au modèle du robot de type (2, 0). En effet on a alors $\dot{z} = \Theta v$

Platitude : Un robot de type (3, 0) admet (z, Θ) comme sorties plates (il y a trois entrées au modèle et donc trois sorties plates, z comptant pour deux sorties plates dans \mathbb{R}). Les équations de platitude sont données par :

$$\begin{cases} z = z \\ \Theta = \Theta \\ v_1 = \mathcal{R}\left(\frac{\dot{z}}{\Theta}\right) \\ v_2 = \mathcal{I}\left(\frac{\dot{z}}{\Theta}\right) \\ \omega = -i \times \frac{\dot{\Theta}}{\Theta} \end{cases} \quad (2.79)$$

Démonstration. La démonstration de la platitude du robot de type (3, 0) est immédiate. \square

2.2.3.3 Robot de type (2, 1)

Modèle cinématique : Le modèle cinématique est décrit par l'équation suivante

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} \cos(\theta + \beta) & 0 & 0 \\ \sin(\theta + \beta) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega_1 \\ \omega_2 \end{pmatrix}, \quad (2.80)$$

où v est la vitesse linéaire du robot, ω_1 sa vitesse angulaire et ω_2 la vitesse angulaire de sa roue orientable centrée. En posant :

$$\begin{cases} z = x + iy \\ \Theta = \exp(i\theta) \\ B = \exp(i\beta) \end{cases}, \quad (2.81)$$

le modèle cinématique d'un robot de type (2, 1) peut être alors être réécrit sous la forme suivante :

$$\begin{pmatrix} \dot{z} \\ \dot{\Theta} \\ \dot{B} \end{pmatrix} = \begin{pmatrix} iB\Theta & 0 & 0 \\ 0 & i\Theta & 0 \\ 0 & 0 & iB \end{pmatrix} \begin{pmatrix} v \\ \omega_1 \\ \omega_2 \end{pmatrix} \quad (2.82)$$

En prenant comme nouvelle entrée $\omega_3 = \omega_1 + \omega_2$ et comme orientation générale $\gamma = \theta + \beta$ ($\Gamma = B\Theta$), le modèle est semblable au modèle du robot de type (2, 0).

Platitude : Un robot de type (2, 1) admet (z, B) comme sorties plates (il y a trois entrées au modèle et donc trois sorties plates, z comptant pour deux sorties plates dans \mathbb{R}). Les équations de platitude sont données par :

$$B = B \quad (2.83)$$

$$z = z \quad (2.84)$$

$$\omega_2 = -i \times \frac{\dot{B}}{B} \quad (2.85)$$

$$v = |\dot{z}| \quad (2.86)$$

$$\Theta = -i \times \frac{\dot{z}}{|\dot{z}|B} \quad (2.87)$$

$$\omega_1 = B \times \frac{\ddot{z}|\dot{z}|^2 B - \dot{z}^2 \bar{\ddot{z}} B + |\dot{z}| \dot{B}}{2|\dot{z}|^2 \dot{z}} \quad (2.88)$$

Démonstration. On a $\dot{z} = iB\Theta v$ et $|iB\Theta| = 1$ donc $v = |\dot{z}| \in \mathbb{R}$. Ainsi, $\Theta = \frac{\dot{z}}{iB|\dot{z}|}$. Par ailleurs,

$$\frac{d|u|}{dt} = \frac{\dot{u}\bar{u} + u\dot{\bar{u}}}{2|u|},$$

que l'on utilise pour calculer $\frac{d|\dot{z}|}{dt}$ ce qui permet de calculer $\omega_1 = -i \frac{\dot{\Theta}}{\Theta}$:

$$\omega_1 = -i \times \frac{\ddot{z}|\dot{z}|B - \dot{z} \left[\frac{\dot{z}\bar{\ddot{z}} + \ddot{z}\bar{\dot{z}}}{2|\dot{z}|} \times B + |\dot{z}| \dot{B} \right]}{|\dot{z}|^2} \times \frac{|\dot{z}| iB}{\dot{z}} \quad (2.89)$$

$$= B \times \frac{2\ddot{z}|\dot{z}|^2 B - |\dot{z}|^2 \ddot{z} B - \dot{z}^2 \bar{\ddot{z}} B + |\dot{z}| \dot{B}}{2|\dot{z}|^2 \dot{z}} \quad (2.90)$$

$$= B \times \frac{\ddot{z}|\dot{z}|^2 B - \dot{z}^2 \bar{\ddot{z}} B + |\dot{z}| \dot{B}}{2|\dot{z}|^2 \dot{z}} \quad (2.91)$$

□

2.2.3.4 Robot de type (1, 1)

Modèle cinématique : Le modèle cinématique peut être décrit par les équations suivantes :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} L \cos \theta \sin \beta & 0 \\ L \sin \theta \sin \beta & 0 \\ \cos \beta c_3 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}, \quad (2.92)$$

où v est la vitesse linéaire du robot et ω est sa vitesse angulaire. En posant

$$\begin{cases} z = x + iy \\ \Theta = \exp(i\theta) \\ B = \exp(i\beta) \end{cases}, \quad (2.93)$$

le modèle cinématique d'un robot de type (1, 1) peut être alors être réécrit sous la forme suivante :

$$\dot{z} = iL\Theta\mathcal{I}(B)v, \quad (2.94)$$

$$\dot{\Theta} = i\Theta\mathcal{R}(B)v, \quad (2.95)$$

$$\dot{B} = i\omega B. \quad (2.96)$$

$$\begin{pmatrix} \dot{z} \\ \dot{\Theta} \\ \dot{B} \end{pmatrix} = \begin{pmatrix} iL\Theta\mathcal{I}(B) & 0 \\ i\Theta\mathcal{R}(B) & 0 \\ 0 & iB \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}. \quad (2.97)$$

Platitude : Un robot de type (1, 1) admet (z) comme sorties plates (il y a deux entrées au modèle et donc deux sorties plates, z comptant pour deux sorties plates dans \mathbb{R}). Les équations de platitude sont données par :

$$z = z \quad (2.98)$$

$$\Theta = \frac{\dot{z}}{|\dot{z}|} \quad (2.99)$$

$$B = \frac{i\frac{|\dot{z}|}{L} + \left| \frac{\ddot{z}|\dot{z}|^2 - \dot{z}^2\ddot{z}}{2|\dot{z}|^3} \right|}{\left| i\frac{|\dot{z}|}{L} + \left| \frac{\ddot{z}|\dot{z}|^2 - \dot{z}^2\ddot{z}}{2|\dot{z}|^3} \right| \right|} \quad (2.100)$$

$$u = \left| i\frac{|\dot{z}|}{L} + \left| \frac{\ddot{z}|\dot{z}|^2 - \dot{z}^2\ddot{z}}{2|\dot{z}|^3} \right| \right| \quad (2.101)$$

$$\omega = |\dot{B}| \quad (2.102)$$

Démonstration. En effet,

$$|\dot{z}| = Lv\mathcal{I}(B) \quad (2.103)$$

$$|\dot{\Theta}| = v\mathcal{R}(B) \quad (2.104)$$

$$i \times \frac{|\dot{z}|}{L} + |\dot{\Theta}| = vB \quad (2.105)$$

$$(2.106)$$

Ce qui conduit à ($|B| = 1$) :

$$B = \frac{i \times \frac{|\dot{z}|}{L} + |\dot{\Theta}|}{\left| i \times \frac{|\dot{z}|}{L} + |\dot{\Theta}| \right|} \quad (2.107)$$

$$v = \left| i \times \frac{|\dot{z}|}{L} + |\dot{\Theta}| \right| \quad (2.108)$$

Enfin

$$|\dot{\Theta}| = \left| \frac{\dot{z}|z|^2 - \dot{z}^2\bar{z}}{2|z|^3} \right| \quad (2.109)$$

□

2.2.4 Dérivation numérique

Nous avons vu dans la sous-section 2.2.2.4 que la notion d'observabilité est extrêmement liée à la notion de dérivation numérique. Nous allons présenter ici un dérivateur numérique en environnement bruité.

2.2.4.1 Introduction et état de l'art

L'objectif de la différentiation numérique est l'estimation numérique des dérivées d'une fonction inconnue pour laquelle nous avons des données mesurées discrètes et probablement bruitées. Ce problème a été étudié *via* différentes approches :

- Une approche à base d'observateurs dans le domaine du contrôle [24, 68, 82, 38, 37],
- Une approche par filtre numérique [21, 122, 146],
- Une approche utilisant les équations intégrales de Volterra du premier type [23, 58],
- Une approche par identification [60, 151].

La principale difficulté de ces algorithmes de différentiation numérique est qu'une petite erreur sur les données mesurées peut introduire une grande erreur dans l'estimation de la dérivée. C'est pourquoi de nombreuses méthodes numériques ont été développées pour obtenir des algorithmes stables plus ou moins sensibles au bruit additif. Il est possible de classer ces méthodes dans l'une des huit principales catégories suivantes :

- Les méthodes des différences finies [73, 124, 121, 126],
- Les méthodes de type Savitzky Golay [132, 59, 5, 38],
- Les méthodes de différentiation par vaguelettes [137, 111, 138, 81, 37],
- Les méthodes utilisant une transformée de Fourier [54, 39, 155, 119, 120, 71],
- Les méthodes de suite régularisante [62, 105, 106],
- Les méthodes de régularisation de Tikhonov [30, 61, 107, 154, 150],
- Les méthodes algébriques [102, 101, 87, 84, 86, 85],
- Les méthodes de différentiation par intégration [77, 127, 152],

Les techniques d'estimation de paramètres par méthode algébrique pour les systèmes linéaires ([48, 51]) ont été étendues à une large variété de problèmes dans le cadre du traitement de signal, (voir [47, 45, 100, 109, 110, 143, 144, 145, 88]). Notons, que ces méthodes, algébriques et non asymptotiques, présentent de bonnes propriétés de robustesse vis à vis du bruit et ce sans qu'il soit nécessaire de connaître les propriétés statistiques de ce bruit (voir [43, 44] pour plus de détails théoriques). Cette robustesse vis-à-vis du bruit a été d'ailleurs confirmée par de nombreuses simulations et expérimentations. Ces techniques peuvent aussi être utilisées pour construire des algorithmes numériques de dérivation proposant les mêmes propriétés (voir par exemple [102, 101, 87, 84, 86, 85]). Elles ont déjà été employées dans [50, 49, 6] pour l'estimation d'état et seront mises à profit dans ce travail pour la localisation de robots mobiles. Nous allons commencer par donner

un exemple introductif avant de donner la forme générale d'un dérivateur d'ordre n , dans sa version continue puis dans sa version discrète.

2.2.4.2 Un exemple introductif

Une des idées les plus répandues pour l'estimation de dérivées est d'essayer d'effectuer une régression du signal mesuré par un polynôme. Par exemple, [2, 18] utilisent une méthode d'interpolation alors que [59, 132] utilise une méthode par moindres carrés. Une fois la régression effectuée il est très simple d'estimer la dérivée du signal mesuré à n'importe quel ordre (à condition qu'il ne dépasse pas l'ordre du polynôme obtenu par régression) car cette dérivée est directement liée aux coefficients du polynôme obtenu.

Cette approche, consistant à estimer la dérivée à partir des coefficients d'un polynôme, a été utilisée dans [101, 102], où les coefficients sont estimés en utilisant une méthode algébrique (par une technique d'élimination assez proche de celle introduite dans [65]).

Pour faciliter la compréhension de ces travaux, nous allons commencer par présenter un exemple introductif :

Prenons $p_1(t) = a_0 + a_1 t$ un polynôme du premier ordre sur \mathbb{R}^+ , où a_0 et a_1 sont inconnues. Nous allons calculer la dérivée première de p_1 qui est en fait le coefficient a_1 . Dans cette optique nous appliquons une méthode d'élimination dans le domaine opérationnel. En appliquant la transformée de Laplace à t^α , $\alpha = 0$ et $\alpha = 1$, nous obtenons $\hat{p}_1 = \frac{a_0}{s} + \frac{a_1}{s^2}$, où \hat{p}_1 est la transformée de Laplace de p_1 . Multiplions ensuite chaque membre de l'équation par s , nous obtenons alors $s\hat{p}_1 = a_0 + \frac{a_1}{s}$. Il est alors possible d'annihiler le coefficient a_0 en dérivant par rapport à s la dernière équation. Ce qui donne :

$$s\hat{p}_1^{(1)} + \hat{p}_1 = -\frac{1}{s^2}a_1, \text{ avec } \hat{p}_1^{(1)} = \frac{d\hat{p}_1}{ds}. \quad (2.110)$$

Finalement il ne reste plus que a_1 et \hat{p}_1 dans (2.110). Il faut donc maintenant retourner dans le domaine temporel pour calculer a_1 en utilisant les connaissances que nous avons de p_1 . Cependant, comme la transformée de Laplace inverse de $s\hat{p}_1^{(1)}$ contient la dérivée de p_1 qui est inconnue, nous allons d'abord multiplier chaque membre de l'équation (2.110) par s^{-2} . Pour repasser dans le domaine temporelle nous avons besoin des transformée de Laplace suivante :

Formule de dérivation :

$$\mathcal{L}\{t^n f(t)\}(s) = (-1)^n \frac{d^n}{ds^n} \{\hat{f}(s)\}, \quad (2.111)$$

L'intégral de Riemann-Liouville d'ordre $\alpha \in \mathbb{R}_+$ d'une fonction réelle $f : \mathbb{R} \rightarrow \mathbb{R}$ est définie par

$$J^\alpha f(t) := \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau. \quad (2.112)$$

Pour tout $\alpha \in \mathbb{R}_+$,

$$\mathcal{L}\{t^{\alpha-1}\}(s) = \frac{\Gamma(\alpha)}{s^\alpha}, \quad (2.113)$$

$$\mathcal{L}\{J^\alpha f(t)\}(s) = \frac{1}{s^\alpha} \hat{f}(s). \quad (2.114)$$

De plus en combinant (2.111) et (2.114),

$$\mathcal{L}^{-1} \left\{ \frac{1}{s^\alpha} \frac{d^n \hat{f}(s)}{ds^n} \right\} (t) = \frac{(-1)^n}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \tau^n f(\tau) d\tau. \quad (2.115)$$

Rappelons que $\forall n \in \mathbb{N}, \Gamma(n+1) = n!$
 En utilisant (2.115), nous obtenons

$$\frac{1}{s} \hat{p}_1^{(1)} = - \int_0^t \tau \hat{p}_1^{(1)}(\tau) d\tau$$

En utilisant (2.114), nous obtenons

$$\frac{1}{s^2} \hat{p}_1^{(1)} = \int_0^t (t - \tau) \hat{p}_1^{(1)}(\tau) d\tau$$

Nous avons donc

$$\frac{1}{s} \hat{p}_1^{(1)} + \frac{1}{s^2} \hat{p}_1^{(1)} = \int_0^t (t - 2\tau) \hat{p}_1^{(1)}(\tau) d\tau$$

En utilisant (2.113), avec $\alpha = 4$, nous obtenons

$$\mathcal{L}\{t^3\}(s) = \frac{3!}{s^4}$$

et donc $\mathcal{L}^{-1}\{-s^{-4}\} = -\frac{t^3}{3!}$

En fusionnant l'ensemble nous obtenons alors :

$$a_1 = \frac{3!}{t^3} \int_0^t (2\tau - t) p_1(\tau) d\tau, \quad t > 0. \quad (2.116)$$

Dans les calculs précédents nous avons utilisé l'opérateur différentiel suivant :

$$\Pi_{0,0} = \frac{1}{s^2} \cdot \frac{d}{ds} \cdot s, \quad (2.117)$$

qui nous a permis d'annihiler a_0 et de calculer a_1 par une intégrale. Cet opérateur différentiel est donc un annihilateur que nous pouvons appeler *annihilateur intégral*. Cette méthode (de différentiation) peut être appelée une méthode de *différentiation par intégration*. L'un des avantages d'une telle méthode est qu'une petite fenêtre temporelle $[0, t]$ est suffisante pour obtenir une valeur robuste de a_1 .

2.2.4.3 Annihilateur - Estimateur

L'extension aux fonctions polynomiales d'ordre supérieur est directe. Pour l'estimation de dérivées jusqu'à un certain ordre fini d'une fonction suffisamment lisse donnée, il suffit de prendre une série de Taylor tronquée dans une fenêtre temporelle donnée. Nous appliquons alors des calculs similaires à ceux présentés dans l'exemple introductif de cette partie. Qui plus est, en utilisant une fenêtre glissante, il est possible d'estimer la dérivée de la fonction pour n'importe quel échantillon du signal mesuré. Prenons x une fonction analytique réelle définie sur un intervalle de temps ouvert $I \subset \mathbb{R}^+$. Prenons $n \in \mathbb{N}$, nous allons estimer la dérivée d'ordre n de x . Pour tout $t_0 \in I$, prenons la série de Taylor tronquée à l'ordre N ($N \geq n$) de x autour de t_0

$$x_N(t + t_0) = \sum_{i=0}^N \frac{t^i}{i!} x^{(i)}(t_0). \quad (2.118)$$

Dans cette série de Taylor, nous voulons calculer $x^{(n)}(t_0)$. En appliquant la transformée de Laplace à (2.118) nous obtenons :

$$\hat{x}_N(s) = \sum_{i=0}^N s^{-(i+1)} x^{(i)}(t_0),$$

où $\hat{x}_N(s)$ est la transformée de Laplace de $x_N(t)$ avec $s \in \mathbb{C}$. L'objectif de l'étape suivante est de construire un annihilateur intégral pour annihiler les termes contenant $x^{(i)}(t_0)$ avec $i \neq n$. Il ne reste alors plus qu'à calculer $x^{(n)}(t_0)$ avec une intégrale. L'opérateur différentiel présenté dans [101] :

$$\Pi_{k,\mu}^{N,n} = \frac{1}{s^{N+1+\mu}} \cdot \frac{d^{n+k}}{ds^{n+k}} \cdot \frac{1}{s} \cdot \frac{d^{N-n}}{ds^{N-n}} \cdot s^{N+1}, \quad \text{avec } k, \mu \in \mathbb{N}, \quad (2.119)$$

permet d'obtenir cette estimation ($\Pi_{k,\mu}^{N,n}$ est un annihilateur).

Ensuite, le coefficient $x^{(n)}(t_0)$ dans le membre de droite de (2.118) est conservé en suivant le schéma suivant :

- La multiplication par s^{N+1} de $\hat{x}_N(s)$ donne un polynôme de degré N ,
- Les termes de degré $N - n$ dont $x^{(i)}(t_0)$, $n < i < N$ sont annihilés en appliquant $N - n$ dérivations successives par rapport à s ,
- De manière à préserver le terme $x^{(n)}(t_0)$, le polynôme restant est multiplié par $\frac{1}{s}$,
- Nous appliquons ensuite n dérivations successives par rapport à s pour annihiler les termes $x^{(i)}(t_0)$ avec $0 < i < n$,
- Finalement, la multiplication par $\frac{1}{s^{N+1+\mu}}$ puis l'application de la transformée de Laplace inverse permet de revenir dans le domaine temporel à l'instant t_0 avec une fenêtre temporelle glissante de taille T ($T > 0$ pour que $[t_0, t_0 + T] \subset I$) ne dépendant que de la fonction x ,
- Nous obtenons alors dans le domaine temporel une expression intégrale de $x^{(n)}(t_0)$. En remplaçant x_N par x , l'approximation de $x^{(n)}(t_0)$ donne un estimateur de $x_{t_0}^{(n)}$.

Cet estimateur étant fonction des paramètres T , N , k et μ , nous l'écrivons $\tilde{x}_{t_0+}^{(n)}(k, \mu, T, N)$.

En utilisant une fenêtre d'intégration glissante $[t_0, t_0 + T]$ nous pouvons estimer les dérivées temporelles de x pour tous points $t_0 \in I$ vérifiant la condition $[t_0, t_0 + T] \subset I$. Une des erreurs de cet estimateur provient de la troncature de la série de Taylor de x . Cette erreur est logiquement appelée **erreur de troncature**. Cette erreur peut être réduite en augmentant l'ordre de troncature N . L'estimateur obtenu en prenant $N = n$ est appelé **estimateur minimal**, il est noté $\tilde{x}_{t_0+}^{(n)}(k, \mu, T)$. Cet estimateur est de première importance, car [101, 102] prouvent que l'estimateur $\tilde{x}_{t_0+}^{(n)}(k, \mu, T, N)$ obtenu en utilisant $N > n$ peut être écrit comme une combinaison affine d'un certain nombre d'estimateurs minimaux. Cet estimateur correspond alors à un point de la \mathbb{Q} -enveloppe affine de l'ensemble

$$S_{k,\mu,T,q} = \left\{ \tilde{x}_{t_0+}^{(n)}(k+q, \mu, T), \dots, \tilde{x}_{t_0+}^{(n)}(k, \mu+q, T) \right\} \quad \text{avec } q = N - n. \quad (2.120)$$

Un tel estimateur est appelé **estimateur affine**. Finalement [102] prouve qu'un tel estimateur $\tilde{x}_{t_0+}^{(n)}(k, \mu, T, N)$ ($N \geq n$) peut être aussi obtenu en prenant la q^{me}

($q = N - n \leq n + k$) série des projections orthogonale de Jacobi, tronquée de $x^{(n)}(t_0 + T\xi)$ au point $\xi = 0$:

$$\tilde{x}_{t_0+}^{(n)}(k, \mu, T, N) = \sum_{i=0}^q \frac{\langle P_i^{(\mu+n, k+n)}(\cdot), x^{(n)}(t_0 + T\cdot) \rangle_{\mu+n, k+n}}{\|P_i^{(\mu+n, k+n)}\|_{\mu+n, k+n}^2} P_i^{(\mu+n, k+n)}(\xi), \quad (2.121)$$

avec $\xi = 0$. En prenant $\xi \in [0, 1]$ au lieu de 0, la \mathbb{Q} -enveloppe affine $S_{k, \mu, T, q}$ est étendue à la \mathbb{R} -enveloppe affine. Ainsi, n'importe quel point dans cet ensemble représente un estimateur de $x_{t_0}^{(n)}$.

Caractériser les points qui minimisent une distance donnée à $x^{(n)}(t_0)$ est une question de grande importance. Un choix judicieux a été introduit dans [102] en prenant ξ comme la plus petite racine de $P_{q+1}^{(\mu+n, k+n)}$. Cela revient à prendre la série de Jacobi tronquée à l'ordre $(q+1)$ de $x^{(n)}(t_0 + T\xi)$, qui produit un retard temporel. Nous notons alors ce retard $\tilde{x}_{t_0+}^{(n)}(k, \mu, T, N, \xi)$ et l'erreur d'estimation vient alors de la troncature de la série orthogonale de Jacobi.

L'étude du choix de κ et μ a été faite dans [83]. Cependant, dans ce travail, nous utiliserons $\kappa = \mu = 0$ qui nous permet d'obtenir un retard identique quel que soit l'ordre de dérivation.

2.2.4.4 Version continue de l'estimateur de dérivée

Considérons un signal $\mathbf{y}_m = \mathbf{y} + \varpi_{\mathbf{y}}$ (avec $\varpi_{\mathbf{y}}$ le bruit additif). Nous voulons estimer la dérivée de \mathbf{y} . κ et μ prennent leur valeur dans $[-1, +\infty[$. Le lecteur peut se reporter à [83] pour plus de détails sur les réglages.

La version continue de l'estimateur de la dérivée temporelle n^{th} de la variable \mathbf{y} est donnée par :

$$\hat{\mathbf{y}}_f^{(n)}(t - T\xi; \kappa, \mu, q) = \int_0^1 g_{\kappa, \mu, n, q, \xi}(\tau) \mathbf{y}_m(t - T\tau) d\tau, \quad (2.122)$$

où \mathbf{y}_m est la variable mesurée de \mathbf{y} et

$$\begin{aligned} g_{\kappa, \mu, n, q, \xi}(\tau) &= \frac{(-1)^n}{T^n} Q_{\kappa, \mu, n, q, \xi}(\tau), \\ Q_{\kappa, \mu, n, q, \xi}(\tau) &= w_{\mu, \kappa}(\tau) \sum_{\ell=0}^q C_{\kappa, \mu, n, \ell} P_{\ell}^{(\mu+n, \kappa+n)}(\xi) P_{n+\ell}^{(\mu, \kappa)}(\tau), \\ w_{\mu, \kappa}(\tau) &= (1 - \tau)^{\mu} \tau^{\kappa}, \\ C_{\kappa, \mu, n, \ell} &= \frac{(\mu + \kappa + 2n + 2\ell + 1)\Gamma(\kappa + \mu + 2n + \ell + 1)\Gamma(n + \ell + 1)}{\Gamma(\kappa + n + \ell + 1)\Gamma(\mu + n + \ell + 1)}, \\ P_n^{(\mu, \kappa)}(\tau) &= \sum_{j=0}^n \binom{n + \mu}{j} \binom{n + \kappa}{n - j} (\tau - 1)^{n-j} \tau^j, \end{aligned}$$

où Γ est la fonction gamma, T est la taille de la fenêtre d'intégration, $\mu, \kappa \in]-1, +\infty[$ et $q \in \mathbb{N}, q > 0$, sont des paramètres à choisir (nous prenons ici $\kappa = \mu = 0$ afin d'avoir un retard d'estimation identique quel que soit l'ordre de dérivation souhaité) ξ est connecté au retard (toute technique différentielle causale (même les observateurs) introduit un retard, l'avantage de la présente méthode est de pouvoir l'évaluer et le lier à l'erreur d'estimation, voir [85, 87]).

Remarque 2.9 Cette formule est toujours valide pour $n = 0$ et donne dans ce cas une estimation filtrée de la variable mesurée.

2.2.4.5 Version discrète de l'estimateur de dérivée

L'estimation de la dérivée temporelle à l'ordre n dans un cadre discret peut être obtenue en utilisant une intégration numérique de l'équation (2.122). Cette dérivée est alors la sortie d'un filtre à réponse impulsionnelle finie dont les coefficients sont donnés par :

$$\widehat{\mathbf{y}}^{(n)}(t - T\xi; \kappa, \mu, q) \approx \sum_{j=0}^{M-1} W_j g_j \mathbf{y}_{m, l-j}, \quad (2.123)$$

où $\mathbf{y}_{m, \ell} = \mathbf{y}_m(\ell T_s)$, T_s est la période d'échantillonnage, M est le nombre de coefficients du filtre et est relié à la longueur de la fenêtre d'intégration temporelle par $T = MT_s$. Les coefficients $c_j = W_j g_j$, $j = 0, \dots, M - 1$ caractérisent la réponse impulsionnelle du filtre. Les coefficients W_j dépendent de la méthode d'intégration choisie. Pour la méthode des trapèzes, nous avons : $W_0 = W_M = \frac{1}{2M}$, $W_j = \frac{1}{M}$, $j = 1, \dots, M - 1$. Les coefficients g_j sont donnés par $g_j = g_{\kappa, \mu, n, q, \xi} \left(\frac{j}{M} \right)$, $j = 0, \dots, M$.

CONCLUSION DU CHAPITRE

Dans ce chapitre, nous avons présenté les différents outils qui seront à la base de nos développements. Comme indiqué en préambule, nous souhaitons étudier deux problématiques majeures de la robotique mobile, la localisation et la planification de trajectoire. L'étude des systèmes non-linéaires a été largement traitée par la communauté Automatique. Nous avons décidé d'utiliser les outils développés dans ce cadre pour les adapter à la robotique.

Que ce soit pour la planification de trajectoire ou pour la localisation, il est intéressant de s'appuyer sur le modèle cinématique du robot. Un des gros avantages est qu'un robot mobile à roues quelconques appartient forcément à l'une des cinq classes de robots, chacune des classes étant paramétrée par deux variables, le degré de mobilité et le degré de directionalité eux-mêmes fixés par les contraintes s'appliquant sur les roues supposées roulant sans glisser sur le plan. Pour chacune des cinq classes de robots, il est possible d'exprimer un modèle cinématique générique. Ce modèle cinématique s'exprime comme un ensemble d'équations différentielles.

Dans la deuxième partie de ce chapitre nous avons rappelé les bases mathématiques permettant d'étudier un tel ensemble. Il est possible de définir un ensemble de notions d'algèbre différentielle permettant d'étudier notamment la platitude du système (c'est-à-dire la possibilité de trouver un vecteur de sortie tel que l'ensemble des variables du système (états et entrées) s'exprime en fonction de ce vecteur de sortie et d'un nombre fini de dérivées de ses composantes). Bien qu'introduite dans une perspective de commande, cette notion de platitude est aussi liée à la notion d'observabilité. En effet pour toute sortie plate mesurée par un capteur, le système est observable et il est alors possible en utilisant les équations de la platitude de construire un estimateur d'état et d'entrée pour le système. Cet estimateur d'état s'appuie, nous venons de le voir, sur la dérivée temporelle d'un certain nombre de sorties du système jusqu'à un ordre fini.

En guise de dernier outil de ce chapitre nous avons présenté un dérivateur numérique en environnement bruité qui s'exprime comme un filtre à réponse impulsionnelle finie prenant en entrée un certain nombre d'échantillons du signal à dériver. Dans le prochain chapitre nous allons nous appuyer sur ces outils pour faire une étude de la localisation d'un robot unicycle observant un certain nombre de points d'intérêt.

LOCALISATION D'UN ROBOT DE TYPE UNICYCLE

SOMMAIRE	
3.1	MODÈLE DU ROBOT ET SON MODÈLE DE MESURE 73
3.1.1	Modèle cinématique du robot 73
3.1.2	Capteurs et modèle de mesure 73
3.2	LOCALISABILITÉ 75
3.2.1	Observabilité et Localisabilité 75
3.2.2	Localisabilité dans le cadre géométrique 75
3.2.3	Localisabilité dans le cadre algébrique 75
3.2.4	Exemple 76
3.3	QUELQUES CONDITIONS DE LA LOCALISABILITÉ 76
3.3.1	Présentation des cas d'étude 76
3.3.2	Un point d'intérêt (mesures : azimuth et élévation) 77
3.3.3	Un point d'intérêt et une cible (mesures : azimuths et élévations) 79
3.3.4	Un point d'intérêt avec boussole (mesures : azimuth, élévation et θ) 82
3.3.5	Deux points d'intérêt (mesures : deux élévations) 83
3.3.6	Deux points d'intérêt (mesures : deux azimuths) 86
3.3.7	Deux points d'intérêt (mesures : une élévation et un azimuth) 89
3.3.8	Synthèse des résultats obtenus 91
3.4	SIMULATION ET EXPÉRIMENTATION 91
3.4.1	Paramètres de simulation et cas d'étude 92
3.4.2	Résultats numériques 92
3.4.3	Résultats expérimentaux 93
3.5	FUSION DANS LE CAS MULTI-POINTS D'INTÉRÊT 94
3.5.1	Hypothèse de travail 96
3.5.2	Étude de sensibilité de la solution proposée 96
3.5.3	Algorithme de fusion dans le cas de plusieurs points d'intérêt 97
3.5.4	Résultats en simulation 97
CONCLUSION 99	

DANS ce chapitre, nous allons présenter un nouvel algorithme de localisation pour un robot mobile de type unicycle (c'est-à-dire un robot de type $(2, 0)$ en suivant la classification donnée dans la section 2.1.4.6). Une étude préalable de la

localisabilité d'un robot de type donné utilisant un capteur de type donné est effectuée. Nous présenterons premièrement le modèle du robot et le modèle de mesure utilisé ainsi que cinq hypothèses de mesures pour lesquelles nous allons effectuer une étude exhaustive. Nous supposons utiliser un robot de type $(2, 0)$ équipé d'une caméra lui permettant d'effectuer des mesures angulaires entre lui-même et un certain nombre de points d'intérêt de l'environnement. Une fois l'étude de la localisabilité pour les cinq hypothèses de mesures effectuée, nous construirons un estimateur d'état et de commande lorsque cela est possible. Ces estimateurs d'état seront testés et comparés en simulation avec un filtre de Kalman Étendu et une expérimentation sur un robot réel sera fournie pour l'une des hypothèses capteur se prêtant bien à une implémentation réelle. Le lecteur soucieux d'une lecture rapide de ce chapitre pourra se reporter à la table 3.1 pour connaître les hypothèses de mesures et à la table 3.2 pour trouver une synthèse des résultats obtenus. Finalement nous présenterons une étude de sensibilité dans l'un des cas d'étude. Cette étude sera utilisée pour proposer un algorithme de fusion de données dans un cadre multi-points d'intérêt.

3.1 MODÈLE DU ROBOT ET SON MODÈLE DE MESURE

3.1.1 Modèle cinématique du robot

Nous considérons dans ce chapitre un robot unicycle, décrit par un vecteur d'état de dimension trois composé de sa position (x, y) et son orientation θ (voir la figure 3.1 et dont le modèle cinématique est donné par :

$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{p}, \mathbf{u}) = \begin{pmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{pmatrix}, \quad (3.1)$$

où $\mathbf{p} = (x, y, \theta)^T$ est la posture du robot (dans \mathbb{R}^3), et $\mathbf{u} = (v, \omega)^T$ est le vecteur de contrôle (vitesses linéaire v et angulaire ω).

En utilisant le changement de variable suivant,

$$z = x + iy, \quad (3.2)$$

$$\Theta = \exp(i\theta), \quad (3.3)$$

le modèle cinématique du robot de type $(2, 0)$ dans $\mathbb{C}[X]$ s'écrit :

$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{p}, \mathbf{u}) = \begin{pmatrix} \dot{z} \\ \dot{\Theta} \end{pmatrix} = \begin{pmatrix} v\Theta \\ i\omega\Theta \end{pmatrix}, \quad (3.4)$$

avec $\mathbf{p} = (z, \Theta)^T$, $\mathbf{p} \in \mathbb{C}^2$ la posture du robot, et $\mathbf{u} = (v, \omega)^T$ le vecteur de contrôle. Notons que l'application permettant de passer de l'écriture dans le domaine réel à l'écriture dans le domaine complexe est complètement bijective. Finalement, la platitude d'un tel robot, étudiée dans le chapitre précédent, nous permet d'exprimer également les vitesses en fonction de la posture et de sa dérivée

$$v = |\dot{z}|, \quad (3.5)$$

$$\omega = -i \frac{\dot{\Theta}}{\Theta}. \quad (3.6)$$

3.1.2 Capteurs et modèle de mesure

Le robot mobile utilisé dans les expérimentations est équipé d'une caméra. Un certain nombre de points d'intérêt A_ℓ de coordonnées 3D $(x_{A_\ell}, y_{A_\ell}, h_{A_\ell})^T$ est alors extrait du flux vidéo à fréquence constante. À partir du flux vidéo et une fois les points d'intérêt extraits, il est possible d'obtenir la mesure de l'azimut α_ℓ et de l'élévation β_ℓ . Ces deux mesures sont les angles relatifs entre le robot et le point d'intérêt. Notre modèle de mesure est donc donné par les équations (3.7)-(3.9) en utilisant les relations géométriques entre les positions et les angles (voir la figure 3.1) :

$$\mathbf{y} = \mathbf{h}(\mathbf{p}) = \begin{pmatrix} \alpha_{m,\ell} = \alpha_\ell + \eta_{\alpha_\ell} \\ \beta_{m,\ell} = \beta_\ell + \eta_{\beta_\ell} \end{pmatrix}, \quad (3.7)$$

$$\alpha_\ell = \arctan\left(\frac{y_{r,\ell}}{x_{r,\ell}}\right) - \theta, \quad (3.8)$$

$$\beta_\ell = \arctan\left(\frac{h_{A_\ell}}{\sqrt{(x_{r,\ell})^2 + (y_{r,\ell})^2}}\right), \quad (3.9)$$

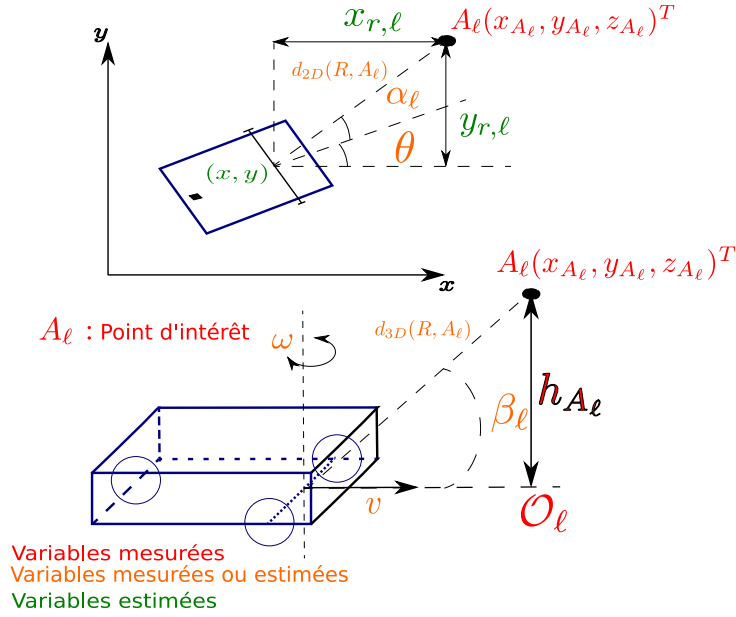


FIGURE 3.1 – Notations pour la localisation

où $\alpha_{m,\ell}$ (respectivement $\beta_{m,\ell}$) est une mesure bruitée de α_ℓ (respectivement β_ℓ), l'azimut relatif (respectivement l'élévation relative) robot-point d'intérêt. Le bruit additif est noté η_{α_ℓ} (respectivement η_{β_ℓ}).

$x_{r,\ell}$ et $y_{r,\ell}$ sont les positions relatives robot-point d'intérêt données par :

$$x_{r,\ell} = x_{A_\ell} - x, \quad y_{r,\ell} = y_{A_\ell} - y. \quad (3.10)$$

Notre étude étant menée dans $\mathbb{C}[X]$, nous allons chercher à réécrire ce modèle de mesure dans cet ensemble. Notons \mathcal{O}_ℓ le projeté du point d'intérêt A_ℓ sur le plan du mouvement, son affixe dans le domaine complexe est notée $z_{A_\ell} = x_{A_\ell} + iy_{A_\ell}$. Nous pouvons alors introduire l'affixe du vecteur reliant le robot et la projection du point d'intérêt dans le plan complexe et la noter $z_{r,\ell} = z_{A_\ell} - z$. Le modèle de mesure s'écrit alors (en utilisant les notations des équations (3.2)-(3.3)) :

$$\mathbf{y} = \mathbf{h}(\mathbf{p}) = \begin{bmatrix} \exp(i\alpha_{m,\ell}) = \exp(i\alpha_\ell) + \eta_{\alpha_\ell} \\ \tan(\beta_{m,\ell}) = \tan(\beta_\ell) + \eta_{\beta_\ell} \end{bmatrix}, \quad \mathbf{p} = (z, \Theta)^T \quad (3.11)$$

$$\Theta \exp(i\alpha_\ell) = \frac{z_{A_\ell} - z}{|z_{A_\ell} - z|} = \frac{z_{r,\ell}}{|z_{r,\ell}|} \quad (3.12)$$

$$\tan \beta_\ell = \frac{h_{A_\ell}}{|z_{A_\ell} - z|} = \frac{h_{A_\ell}}{|z_{r,\ell}|} \quad (3.13)$$

Remarque 3.1 Dans la suite de ce travail, si \mathbf{y}_m est une variable mesurée (et donc bruitée), nous notons \mathbf{y}_f la variable filtrée. Cette variable filtrée est obtenue à partir de la variable mesurée en utilisant un «dérivateur» d'ordre 0. Une variable estimée sera notée avec un chapeau. Ainsi, \hat{v} est une estimation de la vitesse linéaire v du robot. Finalement nous utilisons des sorties auxiliaires notées \mathbf{y}_{aux} qui sont construites par une manipulation algébrique sur les sorties mesurées et sur les paramètres du robot et des points d'intérêt.

3.2 LOCALISABILITÉ

3.2.1 Observabilité et Localisabilité

Nous souhaitons, dans ce travail, donner un cadre commun pour les différentes méthodes de localisation basées sur les points d'intérêt. Nous allons dans cette optique écrire ce problème de localisation comme un problème d'observabilité. Cette écriture nous permet alors de définir une notion qui sera beaucoup utilisée tout au long de ce manuscrit, *la localisabilité*.

Nous avons vu en préambule de ce manuscrit que le modèle d'état d'un robot peut être réécrit sous la forme suivante :

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) + \eta \end{cases} \quad (3.14)$$

Avec $\mathbf{x} \in \mathcal{X}$ le vecteur d'état, $\mathbf{u} \in \mathcal{U}$ le vecteur de contrôle et $\mathbf{y} \in \mathcal{Y}$ le vecteur de mesures. η est un bruit de mesure. $\mathcal{X}, \mathcal{U}, \mathcal{Y}$ sont des variétés différentielles de dimension respective n, m et p .

De manière simplifiée, un système est dit observable s'il est possible de déduire (localement) le vecteur d'état \mathbf{x} à partir des données mesurées (les entrées, \mathbf{u} et les sorties \mathbf{y} ainsi que de leur dérivées successives). De même, un robot est dit localisable s'il est possible de déduire (localement) sa posture (\mathbf{p}) à partir des données mesurées (les entrées, \mathbf{u} et les sorties \mathbf{y} ainsi que de leur dérivées successives). Ces définitions montrent bien le lien existant entre un problème d'observabilité et un problème de localisabilité. En fait, ce lien est d'autant plus fort que dans certains cas (comme pour le robot unicycle par exemple) le vecteur d'état est identique au vecteur de posture. Dans ce cas particulier, l'étude de l'observabilité et celle de la localisabilité se confondent.

Ces différentes remarques nous permettent de donner la définition suivante :

Définition 3.1 (Localisabilité) *Un robot est dit **localisable** si et seulement si son vecteur posture est observable à partir des données mesurées (entrées et/ou sortie).*

Finalement, le problème de localisabilité est un problème **d'observabilité partielle**.

3.2.2 Localisabilité dans le cadre géométrique

Théorème 3.1 *Un robot est **localisable** localement en \mathbf{x} si et seulement si la dimension de la co-distribution d'observabilité, $d\mathcal{O}_{\mathbf{x}}(h) = \mathcal{O}_{\mathbf{x}}(dx, dy, d\theta)^T$ est 3.*

Démonstration. La démonstration du théorème se fait par application directe du résultat de Herman Krener (voir l'annexe A.2 et tout particulièrement le théorème A.1) sur la posture du robot. \square

3.2.3 Localisabilité dans le cadre algébrique

Théorème 3.2 *Un robot est **localisable** si et seulement s'il existe une relation algébrique entre le vecteur posture \mathbf{p} , les entrées \mathbf{u} , les sorties \mathbf{y} et un nombre fini de leurs dérivées temporelles successives.*

Démonstration. La démonstration de ce théorème se fait par application directe du théorème 2.2 sur l'observabilité d'un système non linéaire par l'approche algébrique. \square

Il existe donc un lien fort entre la localisabilité dans un contexte algébrique et la différentiation numérique. Ainsi, s'il est possible de trouver une relation algébrique de la forme :

$$\mathbf{x} = \mathbf{k}(\mathbf{y}, \mathbf{u}, \dot{\mathbf{y}}, \dot{\mathbf{u}}, \dots, \mathbf{y}^{(k)}, \mathbf{u}^{(k)}), k \in \mathbb{N} \quad (3.15)$$

alors, non seulement le robot est localisable par utilisation directe du théorème 3.2 mais qui plus est, la relation (3.15) permet de construire un **estimateur de posture** fonction des données mesurées \mathbf{y} et des entrées \mathbf{u} ainsi qu'un nombre fini de leur dérivées successives. Ces dérivées successives peuvent être calculées numériquement *via* l'approche présentée en sous-section 2.2.4. Finalement, après utilisation de cet algorithme de différentiation numérique, l'estimateur de posture proposé prend en entrées uniquement les données mesurées, le processus de dérivation étant interne à l'estimateur.

3.2.4 Exemple

Prenons un robot non holonome. Son modèle d'état est celui de l'équation, (3.1) dans \mathbb{R} et (3.4) dans \mathbb{C} . Supposons que ce robot mesure sa position $\mathbf{y} = (x, y)$.

Exemple 3.1 (Approche Géométrique)

$$d\mathcal{O}_{\mathbf{x}_0}(\mathbf{h}) = \text{span}\{dx, dy, d(u \cos(\theta)), d(u \sin(\theta))\} = \text{span}\{dx, dy, d\theta\}$$

La dimension de cet ensemble est 3. D'après le théorème 3.1, un robot non holonome mesurant sa position (x, y) est donc localisable au sens géométrique. C'est-à-dire qu'il est possible de reconstruire son orientation avec la connaissance de l'entrée et de la mesure de sa position (sur un intervalle de temps donné).

Exemple 3.2 (Approche Algébrique) *En utilisant les notations complexes nous avons :*

$$\begin{cases} z & = & z \\ \Theta & = & \frac{\dot{z}}{|\dot{z}|} \end{cases} \quad (3.16)$$

Il existe donc bien une équation algébrique reliant le vecteur posture aux sorties mesurées et à un nombre fini $k \in \mathbb{N}$ de leurs dérivées successives (ici $k = 1$). D'après le théorème 3.2, un robot non holonome mesurant sa position (x, y) , est localisable au sens algébrique. Qui plus est, la relation (3.16) peut être utilisée comme un estimateur de posture. En effet, $z = x + iy$ avec x et y les variables mesurées filtrées par un «dérivateur» d'ordre 0 sur les sorties mesurées. De même, Θ peut être estimé en appliquant un dérivateur d'ordre 1 sur ces mêmes sorties mesurées.

3.3 QUELQUES CONDITIONS DE LA LOCALISABILITÉ

3.3.1 Présentation des cas d'étude

Nous allons présenter cinq cas d'étude, chacun des cas étant une combinaison différente des mesures dont les modèles sont donnés par les équations (3.7)-(3.9)

TABLE 3.1 – Présentation des différentes hypothèses capteurs

Nom	Nombre de point d'intérêt	Mesures	Sorties
<u>HM1</u>	1	azimut élévation	$\mathbf{y} = \begin{pmatrix} (3.12) \\ (3.13) \end{pmatrix}$
<u>HM2</u>	1	azimut élévation orientation (compas)	$\mathbf{y} = \begin{pmatrix} (3.12) \\ (3.13) \end{pmatrix}$
<u>HM3</u>	2	2 azimuts	$\mathbf{y} = \begin{pmatrix} (3.13) \\ (3.13) \end{pmatrix}$
<u>HM4</u>	2	2 élévations	$\mathbf{y} = \begin{pmatrix} (3.12) \\ (3.12) \end{pmatrix}$
<u>HM5</u>	2	azimut pour le point d'intérêt 1 élévation pour le point d'intérêt 2	$\mathbf{y} = \begin{pmatrix} (3.12) \\ (3.13) \end{pmatrix}$

dans \mathbb{R} et (3.11)-(3.13) dans \mathbb{C} . Nous noterons chaque hypothèse capteur HM ℓ , avec $\ell = 1 \dots 5$

1. Cas HM1 : le robot observe un point d'intérêt A_1 en mesurant l'azimut α_1 et l'élévation β_1 ,
2. Cas HM2 : le robot observe un point d'intérêt A_1 (mesure de l'azimut α_1 et de l'élévation β_1) et mesure son orientation θ ,
3. Cas HM3 : le robot observe deux points d'intérêt A_1, A_2 et il mesure les deux élévations, β_1 et β_2 ,
4. Cas HM4 : le robot observe deux points d'intérêt A_1, A_2 et il mesure les deux azimuts, α_1 et α_2
5. Cas HM5 : le robot observe deux points d'intérêt A_1, A_2 et il mesure l'azimut pour l'un et l'élévation pour l'autre, β_1 et α_2 .

La table 3.1 présente les cinq hypothèses capteurs et pour chaque cas donne les équations permettant de retrouver le modèle de mesure correspondant.

3.3.2 Un point d'intérêt (mesures : azimut et élévation)

Dans ce cas d'étude, le robot observe un point d'intérêt A_1 et mesure les deux angles relatifs. Le modèle de mesure est donc le suivant (en complexe) :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}| \Theta}, \quad (3.17)$$

$$\mathbf{y}_{m2} = \frac{h_{A_1}}{|z_{r,1}|}. \quad (3.18)$$

3.3.2.1 Étude de la localisabilité

Théorème 3.3 *Un robot de type unicycle n'est pas localisable sous l'hypothèse de mesure HM1. Il est cependant possible de construire un estimateur de commande sous cette hypothèse de mesure.*

Démonstration. La localisabilité du robot équivaut à l'existence d'une relation algébrique de la forme (3.15). Pour cela, construisons à partir des mesures une sortie

auxiliaire :

$$\mathbf{y}_{aux1} = \frac{\mathbf{y}_{m1}}{\mathbf{y}_{m2}} = \frac{z_{r,1}}{\Theta} \quad (3.19)$$

Dérivons cette équation par rapport au temps :

$$\dot{\mathbf{y}}_{aux1} = \frac{\dot{z}_{r,1}\Theta - \dot{\Theta}z_{r,1}}{\Theta^2} \quad (3.20)$$

$$= \frac{-v\Theta^2 - i\omega\Theta z_{r,1}}{\Theta^2} \quad (3.21)$$

$$= -v - i\omega\mathbf{y}_{aux1} \quad (3.22)$$

On a une équation différentielle indépendante qui n'est pas couplée à d'autres variables. Ainsi d^o tr diff $\mathbf{K} < \mathbf{y}_{aux1} > / \mathbf{K} = 1$. Le robot n'est donc pas localisable sous cette hypothèse de mesure.

A défaut d'estimer la posture du robot, il est tout de même possible de reconstruire la vitesse linéaire et la vitesse angulaire du robot en utilisant la partie réelle et la partie imaginaire de l'équation (3.22).

$$\mathcal{R}(\dot{\mathbf{y}}_{aux1}) + i\mathcal{I}(\dot{\mathbf{y}}_{aux1}) = -v + \omega\mathcal{I}(\mathbf{y}_{aux1}) - i\omega\mathcal{R}(\mathbf{y}_{aux1}) \quad (3.23)$$

$$\omega = -\frac{\mathcal{I}(\dot{\mathbf{y}}_{aux1})}{\mathcal{R}(\mathbf{y}_{aux1})} \quad (3.24)$$

$$v = -\mathcal{R}(\dot{\mathbf{y}}_{aux1}) + \mathcal{I}(\dot{\mathbf{y}}_{aux1}) \frac{\mathcal{I}(\mathbf{y}_{aux1})}{\mathcal{R}(\mathbf{y}_{aux1})} \quad (3.25)$$

Les deux dernières équations montrent qu'il est possible de reconstruire les vitesses à partir d'un seul point d'intérêt. \square

Il y a en fait un défaut de localisabilité de 1, il faut donc ajouter une équation algèbro-différentielle non triviale pour pouvoir estimer la posture. Ce sera l'objet d'un nouveau cas d'étude, le cas HM1-BIS dans lequel nous allons lever le défaut de localisabilité du cas HM1

3.3.2.2 Estimateur d'état et des vitesses

Théorème 3.4 *Un robot de type (2,0) observant un unique point d'intérêt et mesurant les angles relatifs entre lui-même et ce point (3.17)-(3.18), peut reconstruire ses commandes en utilisant les estimateurs suivants :*

$$\hat{\omega} = -\frac{\mathcal{I}(\hat{\mathbf{y}}_{aux1})}{\mathcal{R}(\hat{\mathbf{y}}_{aux1})} \quad (3.26)$$

$$\hat{v} = \mathcal{R}(\hat{\mathbf{y}}_{aux1}) - \frac{\mathcal{I}(\hat{\mathbf{y}}_{aux1})\mathcal{I}(\hat{\mathbf{y}}_{aux1})}{\mathcal{R}(\hat{\mathbf{y}}_{aux1})} \quad (3.27)$$

$$\hat{\mathbf{y}}_{aux1} = \frac{\mathbf{y}_{f1}}{\mathbf{y}_{f2}} \quad (3.28)$$

où $\mathbf{y}_{f1}, \mathbf{y}_{f2}$ sont les valeurs filtrées (par un dérivateur d'ordre 0) de \mathbf{y}_{m1} (3.17) et \mathbf{y}_{m2} (3.18).

Démonstration. Utilisons l'expression de \mathbf{y}_{aux1} (3.22) et décomposons \mathbf{y}_{aux1} de la façon suivante $\mathbf{y}_{aux1} = \mathcal{R}(\mathbf{y}_{aux1}) + i\mathcal{I}(\mathbf{y}_{aux1})$ alors :

$$\mathcal{R}(\dot{\mathbf{y}}_{aux1}) + i\mathcal{I}(\dot{\mathbf{y}}_{aux1}) = -v + \omega\mathcal{I}(\mathbf{y}_{aux1}) - i\omega\mathcal{R}(\mathbf{y}_{aux1}) \quad (3.29)$$

Il est donc possible d'estimer u et ω de la façon suivante :

$$\omega = -\frac{\mathcal{I}(\dot{\mathbf{y}}_{aux1})}{\mathcal{R}(\mathbf{y}_{aux1})}, \quad (3.30)$$

$$v = \mathcal{R}(\dot{\mathbf{y}}_{aux1}) - \frac{\mathcal{I}(\dot{\mathbf{y}}_{aux1})\mathcal{I}(\mathbf{y}_{aux1})}{\mathcal{R}(\mathbf{y}_{aux1})}. \quad (3.31)$$

Le résultat final est alors obtenu en remplaçant la sortie auxiliaire et ses dérivées par leur valeurs numériques filtrées. Nous notons alors les variables estimées par un chapeau (voir la remarque 3.1) \square

3.3.3 Un point d'intérêt et une cible (mesures : azimuts et élévations)

Dans l'étude de la localisabilité pour le cas HM1, nous avons vu qu'il n'était pas possible d'estimer la posture du robot. Le défaut de localisabilité étant de un, il faut ajouter une équation algèbro-différentielle non triviale pour pouvoir estimer la posture du robot. Nous décidons d'ajouter une cible fixe, c'est-à-dire un point de l'environnement distinguable par le robot des autres points mais dont on ne connaît pas les coordonnées. Nous allons appeler cette hypothèse capteurs HM1-BIS.

Le cas HM1-BIS est le cas d'un robot qui mesure les angles relatifs entre lui-même et un point d'intérêt et entre lui-même et une cible.

Introduisons un point d'intérêt A_1 de coordonnées connues et une cible A_2 de coordonnées inconnues. Les sorties auxiliaires (les sorties auxiliaires sont dans ce cas équivalentes aux sorties mesurées) sont donc :

$$\mathbf{y}_{m1} = \mathbf{y}_{aux1} = \exp(i\alpha_1) = \frac{z_{r,1}}{\Theta |z_{r,1}|}, \quad (3.32)$$

$$\mathbf{y}_{m2} = \mathbf{y}_{aux2} = \tan(\beta_1) = \frac{h_{A_1}}{|z_{r,1}|}. \quad (3.33)$$

$$\mathbf{y}_{m3} = \mathbf{y}_{aux3} = \exp(i\alpha_2) = \frac{z_{r,2}}{\Theta |z_{r,2}|}, \quad (3.34)$$

$$\mathbf{y}_{m4} = \mathbf{y}_{aux4} = \tan(\beta_2) = \frac{h_{A_2}}{|z_{r,2}|}. \quad (3.35)$$

Théorème 3.5 *Un robot de type (2, 0) observant un point d'intérêt et une cible (point d'intérêt de coordonnées inconnues) et mesurant les angles relatifs entre lui-même et ces points, (3.32)-(3.35), peut reconstruire sa posture, l'altitude de la cible et ses com-*

mandes en utilisant les estimateurs suivants :

$$\hat{v} = \frac{2h_{A_1}\hat{\mathbf{y}}_{aux2}\hat{\mathbf{y}}_{aux1}}{(1 + \hat{\mathbf{y}}_{aux1}^2)\hat{\mathbf{y}}_{aux2}^2}, \quad (3.36)$$

$$\hat{h}_{A_2} = \frac{h_{A_1}\hat{\mathbf{y}}_{aux2}\hat{\mathbf{y}}_{aux1}}{(1 + \hat{\mathbf{y}}_{aux1}^2)\hat{\mathbf{y}}_{aux2}^2} \times \frac{(1 + \hat{\mathbf{y}}_{aux3}^2)\hat{\mathbf{y}}_{aux4}^2}{\hat{\mathbf{y}}_{aux4}\hat{\mathbf{y}}_{aux3}}, \quad (3.37)$$

$$\hat{\Theta}(t) = \Theta(t_0) \frac{h_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t_0)}{\hat{\mathbf{y}}_{aux2}(t_0)} - \hat{h}_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t_0)}{\hat{\mathbf{y}}_{aux4}(t_0)}}{h_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)} - \hat{h}_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t)}{\hat{\mathbf{y}}_{aux4}(t)}}, \quad (3.38)$$

$$\hat{z}(t) = z_{A_1} - h_{A_1} \Theta(t) \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)}, \quad (3.39)$$

$$\hat{\omega} = -i \frac{num}{denum}, \quad (3.40)$$

$$\begin{aligned} num &= h_{A_1}\hat{\mathbf{y}}_{aux4}^2 \left(\hat{\mathbf{y}}_{aux1}\hat{\mathbf{y}}_{aux2} - \hat{\mathbf{y}}_{aux1}\hat{\mathbf{y}}_{aux2} \right) \\ &\quad + \hat{h}_{A_2}\hat{\mathbf{y}}_{aux2}^2 \left(\hat{\mathbf{y}}_{aux3}\hat{\mathbf{y}}_{aux4} - \hat{\mathbf{y}}_{aux3}\hat{\mathbf{y}}_{aux4} \right), \\ denum &= h_{A_1}\hat{\mathbf{y}}_{aux1}\hat{\mathbf{y}}_{aux2}\hat{\mathbf{y}}_{aux4}^2 - \hat{h}_{A_2}\hat{\mathbf{y}}_{aux2}\hat{\mathbf{y}}_{aux3}\hat{\mathbf{y}}_{aux4}, \\ \hat{\mathbf{y}}_{aux\ell} &= y_{f\ell}, \ell = 1, \dots, 4. \end{aligned} \quad (3.41)$$

Démonstration. Commençons par estimer l'altitude de la cible h_{A_2} qui apparaît dans la construction de la sortie auxiliaire (3.35). Calculons la dérivée de \mathbf{y}_{aux2} :

$$\dot{\mathbf{y}}_{aux2} = -h_{A_1} \frac{d|z_{r,1}|}{|z_{r,1}|^2}. \quad (3.42)$$

Nous avons :

$$\frac{d|z_{r,\ell}|}{dt} = \frac{z_{r,\ell}\bar{z}_{r,\ell} + \bar{z}_{r,\ell}\dot{z}_{r,\ell}}{2|z_{r,\ell}|}, \quad (3.43)$$

ce qui pour $\ell = 1$, permet d'écrire en association avec (3.42) :

$$\dot{\mathbf{y}}_{aux2} = \frac{h_{A_1}v}{2|z_{r,1}|^3} (z_{r,1}\Theta^{-1} + \bar{z}_{r,1}\Theta), \quad (3.44)$$

$$\dot{\mathbf{y}}_{aux2} = \frac{v(1 + \mathbf{y}_{aux1}^2)\mathbf{y}_{aux2}^2}{2h_{A_1}\mathbf{y}_{aux1}}, \quad (3.45)$$

$$v = \frac{2h_{A_1}\dot{\mathbf{y}}_{aux2}\mathbf{y}_{aux1}}{(1 + \mathbf{y}_{aux1}^2)\mathbf{y}_{aux2}^2}. \quad (3.46)$$

Cette dernière relation permet, en remplaçant la sortie auxiliaire et ses dérivées par leur valeurs numériques filtrées, d'obtenir l'estimateur de vitesse linéaire, \hat{v} .

En calculant la dérivée temporelle de \mathbf{y}_{aux4} , on obtient une équation similaire à 3.46 conduisant à la relation :

$$h_{A_2} = v \frac{(1 + \mathbf{y}_{aux3}^2)\mathbf{y}_{aux4}^2}{2\dot{\mathbf{y}}_{aux4}\mathbf{y}_{aux3}}, \quad (3.47)$$

$$h_{A_2} = \frac{h_{A_1}\dot{\mathbf{y}}_{aux2}\mathbf{y}_{aux1}}{(1 + \mathbf{y}_{aux1}^2)\mathbf{y}_{aux2}^2} \times \frac{(1 + \mathbf{y}_{aux3}^2)\mathbf{y}_{aux4}^2}{\dot{\mathbf{y}}_{aux4}\mathbf{y}_{aux3}}. \quad (3.48)$$

Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de l'altitude de la cible h_{A_2}

La distance relative parcourue entre deux instants est la même par rapport au point d'intérêt que par rapport à la cible, ainsi :

$$z_{r,1}(t) - z_{r,1}(t_0) = z_{r,2}(t) - z_{r,2}(t_0), \quad (3.49)$$

Comme $z_{r,1}(t) = h_{A_1} \Theta \frac{y_{aux1}(t)}{y_{aux2}(t)}$ et $z_{r,2}(t) = h_{A_2} \Theta \frac{y_{aux3}(t)}{y_{aux4}(t)}$, l'équation (3.49) peut s'écrire :

$$h_{A_1} \left(\Theta(t) \frac{y_{aux1}(t)}{y_{aux2}(t)} - \Theta(t_0) \frac{y_{aux1}(t_0)}{y_{aux2}(t_0)} \right) = h_{A_2} \left(\Theta(t) \frac{y_{aux3}(t)}{y_{aux4}(t)} - \Theta(t_0) \frac{y_{aux3}(t_0)}{y_{aux4}(t_0)} \right),$$

ce qui donne

$$\Theta(t) = \Theta(t_0) \frac{h_{A_1} \frac{y_{aux1}(t_0)}{y_{aux2}(t_0)} - h_{A_2} \frac{y_{aux3}(t_0)}{y_{aux4}(t_0)}}{h_{A_1} \frac{y_{aux1}(t)}{y_{aux2}(t)} - h_{A_2} \frac{y_{aux3}(t)}{y_{aux4}(t)}}. \quad (3.50)$$

$$(3.51)$$

Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de l'orientation du robot $\hat{\Theta}$.

Nous pouvons alors obtenir la position $z(t)$ du robot :

$$z(t) = z_{A_1} - h_{A_1} \Theta \frac{y_{aux1}(t)}{y_{aux2}(t)}, \quad (3.52)$$

où Θ est donné par (3.50) (avec h_{A_2} donné par (3.48)). Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de la position du robot. Il ne reste plus qu'à estimer ω . La dérivée temporelle de l'équation (3.49) donne $\dot{z}_{r,1} = \dot{z}_{r,2}$ ce qui conduit à

$$h_{A_1} \left(\dot{\Theta} \frac{y_{aux1}}{y_{aux2}} + \Theta \frac{\dot{y}_{aux1} y_{aux2} - y_{aux1} \dot{y}_{aux2}}{y_{aux2}^2} \right) = h_{A_2} \left(\dot{\Theta} \frac{y_{aux3}}{y_{aux4}} + \Theta \frac{\dot{y}_{aux3} y_{aux4} - y_{aux3} \dot{y}_{aux4}}{y_{aux4}^2} \right).$$

$$\begin{aligned} \frac{\dot{\Theta}}{\Theta} &= \frac{h_{A_1} \frac{\dot{y}_{aux1} y_{aux2} - y_{aux1} \dot{y}_{aux2}}{y_{aux2}^2} - h_{A_2} \frac{\dot{y}_{aux3} y_{aux4} - y_{aux3} \dot{y}_{aux4}}{y_{aux4}^2}}{h_{A_1} \frac{y_{aux1}}{y_{aux2}} - h_{A_2} \frac{y_{aux3}}{y_{aux4}}}, \\ \frac{\dot{\Theta}}{\Theta} &= \frac{h_{A_1} y_{aux4}^2 (\dot{y}_{aux1} y_{aux2} - y_{aux1} \dot{y}_{aux2}) - h_{A_2} y_{aux2}^2 (\dot{y}_{aux3} y_{aux4} - y_{aux3} \dot{y}_{aux4})}{h_{A_1} y_{aux1} y_{aux2} y_{aux4}^2 - h_{A_2} y_{aux2}^2 y_{aux3} y_{aux4}}, \\ \frac{\dot{\Theta}}{\Theta} &= \frac{h_{A_1} y_{aux4}^2 (y_{aux1} \dot{y}_{aux2} - \dot{y}_{aux1} y_{aux2}) + h_{A_2} y_{aux2}^2 (\dot{y}_{aux3} y_{aux4} - y_{aux3} \dot{y}_{aux4})}{h_{A_1} y_{aux1} y_{aux2} y_{aux4}^2 - h_{A_2} y_{aux2}^2 y_{aux3} y_{aux4}}. \end{aligned}$$

En utilisant l'équation (3.6), nous obtenons finalement

$$\omega = -i \frac{h_{A_1} y_{aux4}^2 (y_{aux1} \dot{y}_{aux2} - \dot{y}_{aux1} y_{aux2}) + h_{A_2} y_{aux2}^2 (\dot{y}_{aux3} y_{aux4} - y_{aux3} \dot{y}_{aux4})}{h_{A_1} y_{aux1} y_{aux2} y_{aux4}^2 - h_{A_2} y_{aux2}^2 y_{aux3} y_{aux4}}. \quad (3.53)$$

Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de la vitesse angulaire du robot, $\hat{\omega}$. \square

3.3.4 Un point d'intérêt avec boussole (mesures : azimut, élévation et θ)

Les travaux de cette sous-section ont fait l'objet de publications dans des conférences internationales [134] [135] Dans ce cas d'étude, noté HM2, le robot observe un point d'intérêt et mesure les deux angles relatifs ainsi que son orientation. Le modèle de mesure est le suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Theta}, \quad (3.54)$$

$$\mathbf{y}_{m2} = \frac{h_{A_1}}{|z_{r,1}|}, \quad (3.55)$$

$$\mathbf{y}_{m3} = \Theta. \quad (3.56)$$

3.3.4.1 Étude de la localisabilité

Théorème 3.6 *Un robot de type unicycle qui mesure les angles relatifs entre lui-même et un point d'intérêt et qui mesure en plus son orientation est localisable. Il est de plus possible d'estimer son vecteur commande (vitesses linéaire et angulaire).*

Démonstration. Cherchons une relation algébrique de la forme (3.15). Pour cela, construisons à partir des mesures les deux sorties auxiliaires suivantes (à condition que $h_{A_1} \neq 0$) :

$$\mathbf{y}_{aux1} = z_{r,1}, \quad (3.57)$$

$$\mathbf{y}_{aux2} = \Theta. \quad (3.58)$$

Dans ce cas la localisabilité est évidente. En effet :

$$z = z_{A_1} - \mathbf{y}_{aux1}, \quad (3.59)$$

$$\Theta = \mathbf{y}_{aux2}. \quad (3.60)$$

où z_{A_1} est le complexe donnant la position de la projection du point d'intérêt dans le plan d'évolution du robot. Nous avons donc trouvé une relation algébrique de la forme (3.15). Le résultat est alors direct par application du théorème 3.2. Pour les vitesses il suffit alors d'utiliser (3.5) et (3.6) \square

3.3.4.2 Estimateur d'état et des vitesses

Théorème 3.7 *Un robot de type unicycle mesurant les angles relatifs entre lui et un point d'intérêt et qui mesure en plus son orientation, (3.54)-(3.56), peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\hat{z} = z_{A_1} - \hat{\mathbf{y}}_{aux1}, \quad (3.61)$$

$$\hat{\Theta} = \hat{\mathbf{y}}_{aux2} \quad (3.62)$$

$$\hat{v} = -\frac{\hat{\mathbf{y}}_{aux1}}{\hat{\mathbf{y}}_{aux2}}, \quad (3.63)$$

$$\hat{\omega} = \frac{-i\hat{\mathbf{y}}_{aux1}}{\hat{\mathbf{y}}_{aux2}}, \quad (3.64)$$

$$\{\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}\} = \left\{ \frac{\mathbf{y}_{f1}\mathbf{y}_{f3}h_{A_1}}{\mathbf{y}_{f2}}, \mathbf{y}_{f3} \right\}. \quad (3.65)$$

Démonstration. À condition que $h_{A_1} \neq 0$, il est possible de construire les sorties auxiliaires y_{aux1}, y_{aux2} données par (3.57)-(3.58). Dans ce cas il est possible de construire de façon évidente :

$$z = z_{A_1} - y_{aux1}, \quad (3.66)$$

$$\Theta = y_{aux2}. \quad (3.67)$$

Ces équations permettent en remplaçant les sorties auxiliaires par leur valeurs numériques filtrées d'obtenir l'estimateur de la posture du robot. Finalement en utilisant les équations (3.5) et (3.6) nous obtenons :

$$v = -\frac{\dot{y}_{aux1}}{y_{aux1}}, \quad (3.68)$$

$$\omega = \frac{-i\dot{y}_{aux2}}{y_{aux2}}. \quad (3.69)$$

Ces deux équations permettent, en remplaçant les sorties auxiliaires par leurs valeurs numériques filtrées, d'obtenir l'estimateur des commandes du robot. \square

3.3.5 Deux points d'intérêt (mesures : deux élévations)

Dans ce cas d'étude, noté HM3, le robot observe deux points d'intérêt et mesure les deux élévations relatives. Le modèle de mesure est le suivant :

$$y_{m1} = \frac{h_{A_1}}{|z_{r,1}|}, \quad (3.70)$$

$$y_{m2} = \frac{h_{A_2}}{|z_{r,2}|}. \quad (3.71)$$

3.3.5.1 Étude de la localisabilité

Théorème 3.8 *Un robot de type unicycle mesurant les élévations entre lui et deux points d'intérêt est localisable au sens fort, c'est à dire qu'il peut reconstruire sa posture mais aussi son vecteur commande.*

Démonstration. Cherchons une relation algébrique de la forme (3.15). Pour cela, construisons à partir des mesures les deux sorties auxiliaires suivantes (à condition que $h_{A_\ell} \neq 0, \ell = 1, 2$) :

$$y_{aux1} = \frac{1}{2}|z_{r,1}|^2, \quad (3.72)$$

$$y_{aux2} = \frac{1}{2}|z_{r,2}|^2. \quad (3.73)$$

Prenons \mathcal{C}_ℓ le cercle centré en \mathcal{O}_ℓ de rayon $r_\ell = \sqrt{2y_{aux\ell}}, \ell = 1, 2$. Nous avons $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$ à condition que :

1. C1 : $|z_{A_1} - z_{A_2}| < r_1 + r_2$
2. C2 : $|z_{A_1} - z_{A_2}| > |r_1 - r_2|$

Rappelons que z_{A_i} est le complexe donnant la position de la projection du point d'intérêt A_i dans le plan d'évolution du robot.

Notons qu'il est impossible que les deux sorties auxiliaires soient simultanément nulles. Ainsi il ne reste que deux cas à étudier :

1. $|z_{r,\ell}| \neq 0, \ell = 1, 2$. Dans ce cas les deux contraintes précédentes sont satisfaites et le robot est localisable,
2. Seulement l'un des deux $|z_{r,\ell}|$ est nul et dans ce cas le robot est situé en \mathcal{O}_ℓ tel que $|z_{r,\ell}| = 0$.

Il ne reste plus qu'à estimer Θ . Comme $\mathbf{y}_{aux\ell} = -v(\cos \theta x_{r,\ell} + \sin \theta y_{r,\ell})$ nous avons :

$$\dot{\mathbf{y}}_{aux\ell} = -v\Theta^{-1}z_{r,\ell}, \ell = 1, 2$$

Il est facile d'estimer Θ à partir de ces deux relations (valable pour $\ell = 1, 2$). Qui plus est ces deux relations peuvent être utilisées pour supprimer l'une des deux possibilités résultantes de l'intersection des deux cercles. Nous avons donc trouvé une relation algébrique de la forme (3.15). Le résultat est alors direct par application du théorème 3.2 \square

3.3.5.2 Estimateur d'état et des vitesses

Théorème 3.9 *Un robot unicycle mesurant les élévations relatives entre lui-même et deux points d'intérêt (Équations (3.70)-(3.71) peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\hat{z} = z_{A_1} + z_{A_{12}} \left(f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm \sqrt{r_1^2 - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})} \right), \quad (3.74)$$

$$\hat{\Theta} = \frac{\hat{z}}{|\hat{z}|}. \quad (3.75)$$

$$\hat{v} = |\hat{z}|, \quad (3.76)$$

$$\hat{\omega} = -\frac{i}{2} \left(\frac{\hat{z}}{|\hat{z}|} - \frac{\hat{z}\bar{\hat{z}}}{|\hat{z}|^2} \right). \quad (3.77)$$

$$\hat{z} = z_{A_{12}} \left(\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm i \frac{\hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})}{\sqrt{2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})}} \right),$$

$$\hat{z} = z_{A_{12}} \left(\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm \left(+i c_{A_{12}} \left(\frac{num}{denum} \right) - i c_{A_{12}} \left(\frac{(\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2})^2}{denum} \right) \right) \right),$$

$$num = (2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})) \left(\hat{\mathbf{y}}_{aux1} - \frac{\hat{\mathbf{y}}_{aux1}^2}{2\hat{\mathbf{y}}_{aux1}} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \right)$$

$$denum = (2\hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}))^{\frac{3}{2}},$$

$$f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}} + \frac{d_{A_{12}}}{2}, \quad (3.78)$$

$$\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}}, \quad (3.79)$$

$$\ddot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}}, \quad (3.80)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|, \quad (3.81)$$

$$\mathbf{y}_{aux\ell} = \frac{1}{2} \left(\frac{h_{A\ell}}{\mathbf{y}_{f\ell}} \right)^2, \ell = 1, 2. \quad (3.82)$$

Démonstration. En utilisant l'hypothèse capteur HM3, quand $h_{A\ell} \neq 0, \ell = 1, 2$, il est possible de prendre les sorties auxiliaires suivantes (3.72) et (3.73) :

$$\begin{aligned} \mathbf{y}_{aux1} &= \frac{1}{2} |z_{r,1}|^2, \\ \mathbf{y}_{aux2} &= \frac{1}{2} |z_{r,2}|^2. \end{aligned}$$

En utilisant l'équation (A.16) donnée en annexe A.4, où les cercles sont de rayon $r_\ell = \sqrt{2\mathbf{y}_\ell}, \ell = 1, 2$, nous obtenons :

$$z = z_{A1} + z_{A12} \left(f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}) \pm \sqrt{r_1^2 - f^2(\mathbf{y}_{aux1}, \mathbf{y}_{aux2})} \right), \quad (3.83)$$

$$f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}) = \frac{\mathbf{y}_{aux1} - \mathbf{y}_{aux2}}{d_{A12}} + \frac{d_{A12}}{2}, \quad (3.84)$$

Ces deux équations permettent, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de la position du robot.

Dans ce cas, sous l'hypothèse de mesure HM3, le rayon des deux cercles ne dépendent que des mesures. L'estimation des dérivées temporelles première et seconde de z par l'équation (A.23) et l'équation (A.24) permettent l'estimation de la vitesse linéaire v (Eq. (3.5) et la vitesse angulaire ω (Eq. (3.6)).

$$\begin{aligned} \dot{z} &= z_{A12} \left(\dot{f}(\dot{\mathbf{y}}_{aux1}, \dot{\mathbf{y}}_{aux2}) \pm i \frac{\dot{\mathbf{y}}_{aux1} - f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}) \dot{f}(\dot{\mathbf{y}}_{aux1}, \dot{\mathbf{y}}_{aux2})}{\sqrt{2\mathbf{y}_{aux1} - f^2(\mathbf{y}_{aux1}, \mathbf{y}_{aux2})}} \right), \\ \ddot{z} &= z_{A12} \left(\ddot{f}(\ddot{\mathbf{y}}_{aux1}, \ddot{\mathbf{y}}_{aux2}) \right) \pm \left(\left(+iz_{A12} \frac{num}{denum} \right) - iz_{A12} \frac{(\dot{\mathbf{y}}_{aux1} - \dot{\mathbf{y}}_{aux2})^2}{denum} \right) \end{aligned}$$

$$\begin{aligned} num &= (2\mathbf{y}_{aux1} - f^2(\mathbf{y}_{aux1}, \mathbf{y}_{aux2})) \left(\ddot{\mathbf{y}}_{aux1} - \frac{\dot{\mathbf{y}}_{aux1}^2}{2\mathbf{y}_{aux1}} - f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}) \ddot{f}(\ddot{\mathbf{y}}_{aux1}, \ddot{\mathbf{y}}_{aux2}) \right) \\ denum &= (2\mathbf{y}_{aux1} - f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2})^2)^{\frac{3}{2}} \end{aligned}$$

$$\dot{f}(\dot{\mathbf{y}}_{aux1}, \dot{\mathbf{y}}_{aux2}) = \frac{\dot{\mathbf{y}}_{aux1} - \dot{\mathbf{y}}_{aux2}}{d_{A12}}, \quad (3.85)$$

$$\ddot{f}(\ddot{\mathbf{y}}_{aux1}, \ddot{\mathbf{y}}_{aux2}) = \frac{\ddot{\mathbf{y}}_{aux1} - \ddot{\mathbf{y}}_{aux2}}{d_{A12}}. \quad (3.86)$$

Nous pouvons donc calculer

$$v = |\dot{z}|. \quad (3.87)$$

Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de la vitesse linéaire du robot. Enfin en combinant les équations précédentes avec :

$$\Theta = \frac{\dot{z}}{|\dot{z}|}. \quad (3.88)$$

on estime Θ .

Cette dernière équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de l'orientation du robot.

En utilisant (3.6), puisque l'on a :

$$\frac{\dot{\Theta}}{\Theta} = \left(\frac{\ddot{z}|z|^2 - \dot{z}^2\bar{z}}{2|\dot{z}|^3} \right) \times \left(\frac{|\dot{z}|}{\dot{z}} \right), \quad (3.89)$$

nous obtenons ω :

$$\omega = -\frac{i}{2} \left(\frac{\ddot{z}}{\dot{z}} - \frac{\dot{z}\bar{\ddot{z}}}{|\dot{z}|^2} \right). \quad (3.90)$$

Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de la vitesse angulaire du robot. \square

3.3.6 Deux points d'intérêt (mesures : deux azimuts)

Dans ce cas d'étude, noté HM4, le robot observe deux points d'intérêt et mesure les deux azimuts relatifs. Le modèle de mesure est le suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Theta}, \quad (3.91)$$

$$\mathbf{y}_{m2} = \frac{z_{r,2}}{|z_{r,2}|\Theta}. \quad (3.92)$$

3.3.6.1 Étude de la localisabilité

Théorème 3.10 *Un robot de type unicycle qui mesure les azimuts entre lui-même et deux points d'intérêt A_1 et A_2 est localisable. Il peut donc reconstruire sa posture, cependant cette reconstruction dépend du vecteur d'entrée qui ne peut donc pas être reconstruit.*

Démonstration. Cherchons une relation algébrique de la forme (3.15). Pour cela, construisons à partir des mesures les deux sorties auxiliaires suivantes :

$$\Theta \mathbf{y}_{aux\ell} = \frac{z_{r,\ell}}{|z_{r,\ell}|}, \quad \ell = 1, 2 \quad (3.93)$$

La dérivation temporelle de ces deux équations donne

$$\dot{\Theta} \mathbf{y}_{aux\ell} + \Theta \dot{\mathbf{y}}_{aux\ell} = \frac{\dot{z}_{r,\ell} |z_{r,\ell}| - z_{r,\ell} \frac{d|z_{r,\ell}|}{dt}}{|z_{r,\ell}|^2}, \quad \ell = 1, 2. \quad (3.94)$$

Puisque

$$\frac{d|z_{r,\ell}|}{dt} = \frac{z_{r,\ell}\bar{\dot{z}}_{r,\ell} + \bar{z}_{r,\ell}\dot{z}_{r,\ell}}{2|z_{r,\ell}|}, \quad (3.95)$$

avec $\dot{z}_{r,\ell} = -\dot{z} = -v\Theta$ et $\bar{\dot{z}}_{r,\ell} = -v\bar{\Theta} = -v\Theta^{-1}$, la relation algébrique (3.94)

donne (en utilisant (3.4)) :

$$\Theta(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell}) = \frac{-v\Theta|z_{r,\ell}| - z_{r,\ell} \left[\frac{-z_{r,\ell}v\Theta^{-1} - \bar{z}_{r,\ell}v\Theta}{2|z_{r,\ell}|} \right]}{|z_{r,\ell}|^2}, \quad (3.96)$$

$$(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell}) = -v \frac{|z_{r,\ell}|^2 - z_{r,\ell}^2 \Theta^{-2}}{2|z_{r,\ell}|^3}, \quad (3.97)$$

$$(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell}) = -v \frac{1 - \mathbf{y}_{aux\ell}^2}{2|z_{r,\ell}|}. \quad (3.98)$$

Nous obtenons finalement

$$|z_{r,\ell}| = \frac{-v(1 - \mathbf{y}_{aux\ell}^2)}{2(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell})}. \quad (3.99)$$

Prenons \mathcal{C}_ℓ le cercle centré en z_{A_ℓ} de rayon $|z_{r,\ell}|$ donné par l'équation (3.99), pour $\ell = 1, 2$. Nous avons $\mathcal{C}_{aux1} \cap \mathcal{C}_{aux2} \neq \emptyset$, dès que :

1. C1 : $|z_{A_1} - z_{A_2}| < |z_{r,1}| + |z_{r,2}|$
2. C2 : $|z_{A_1} - z_{A_2}| > ||z_{r,1}| - |z_{r,2}||$.

Comme les deux rayons $|z_{r,\ell}|$ ne peuvent pas être simultanément nuls, il reste deux cas à distinguer pour satisfaire les deux contraintes C1 et C2 :

1. $|z_{r,\ell}| \neq 0, \ell = 1, 2$: dans ce cas les deux conditions C1-C2 sont satisfaites. Clairement, nous avons $|z_{r,\ell}| \neq 0, \ell = 1, 2$ si et seulement si $v \neq 0$ et si le robot n'est pas aligné avec la projection des deux points d'intérêt dans le plan du mouvement ($\mathbf{y}_{aux1} \neq 1$ et $\mathbf{y}_{aux2} \neq 1$). Si toutes ces conditions sont satisfaites, alors $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$ et z est l'un des deux points d'intersection des deux cercles \mathcal{C}_1 et \mathcal{C}_2 ,
2. Si seulement l'un des $|z_{r,\ell}|$ est nul alors le robot est situé en A_ℓ , avec ℓ tel que $|z_{r,\ell}| = 0$.

Il ne reste plus qu'à estimer Θ (et donc θ) qui peut être obtenue directement en utilisant (3.93). Qui plus est, ces deux relations peuvent être utilisées pour éliminer un des deux points d'intersection des deux cercles.

Nous avons donc trouvé une relation algébrique de la forme (3.15). Le résultat est alors direct par application du théorème 3.2

Il n'est pas possible dans ce cas d'obtenir les vitesses linéaire et angulaire du robot, v, ω à partir des mesures. En effet, le rayon du cercle dépend des mesures mais aussi de la vitesse linéaire et de la vitesse angulaire. Le calcul des dérivées première et seconde de z nécessaires au calcul de v et ω (voir les équations (3.5) et (3.6)) amènent à une relation de la forme :

$$\dot{z} = g_z(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}) \quad (3.100)$$

$$v = g_v(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}) \quad (3.101)$$

$$\ddot{z} = g_{\ddot{z}}(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}, \ddot{v}, \ddot{\omega}, \ddot{\mathbf{y}}_{fl}) \quad (3.102)$$

$$\omega = g_\omega(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}, \ddot{v}, \ddot{\omega}, \ddot{\mathbf{y}}_{fl}) \quad (3.103)$$

dans lesquelles les variables inconnues apparaissent. Il n'est donc pas possible pour les vitesses de trouver une relation de la forme (3.15) et il n'est donc pas possible d'estimer les vitesses sous l'hypothèse de mesure HM4.

□

3.3.6.2 Estimateur d'état

Théorème 3.11 *Un robot de type unicycle qui mesure les azimuts relatifs entre lui-même et deux points d'intérêt, (3.91)-(3.92) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned} \hat{z} &= z_{A_1} + z_{A_{12}} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \\ &\pm z_{A_{12}} \left(i \sqrt{\frac{(1 - \hat{\mathbf{y}}_{aux1}^2)^2}{(i\omega \hat{\mathbf{y}}_{aux1} + \hat{\mathbf{y}}_{aux1})^2} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})} \right), \end{aligned} \quad (3.104)$$

$$\hat{\Theta} = \frac{z_{A_1} - z}{|z_{A_1} - z| \times \hat{\mathbf{y}}_{aux1}}, \quad (3.105)$$

$$\begin{aligned} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) &= \frac{v^2}{8d_{A_{12}}} \left(\frac{(1 - \hat{\mathbf{y}}_{aux1}^2)^2}{(i\omega \hat{\mathbf{y}}_{aux1} + \hat{\mathbf{y}}_{aux1})^2} - \frac{(1 - \hat{\mathbf{y}}_{aux2}^2)^2}{(i\omega \hat{\mathbf{y}}_{aux2} + \hat{\mathbf{y}}_{aux2})^2} \right) \\ &+ \frac{d_{A_{12}}}{2}. \end{aligned} \quad (3.106)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (3.107)$$

Démonstration. En utilisant l'hypothèse de mesure **HM4**, quand $v \neq 0$ il est possible de prendre les sorties auxiliaires suivantes (3.93) :

$$\begin{aligned} \mathbf{y}_1 &= \frac{z_{r,1}}{\Theta |z_{r,1}|}, \\ \mathbf{y}_2 &= \frac{z_{r,2}}{\Theta |z_{r,2}|}. \end{aligned}$$

En utilisant l'équation (A.16) donnée en annexe A.4, où le rayon des cercles est donné par $r_\ell = \frac{-v(1 - \mathbf{y}_{aux\ell}^2)}{2(i\omega \mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell})}$, $\ell = 1, 2$, nous obtenons :

$$\begin{aligned} z &= z_{A_1} + z_{A_{12}} (f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}, \dot{\mathbf{y}}_{aux1}, \dot{\mathbf{y}}_{aux2}) \\ &\pm z_{A_{12}} \left(i \sqrt{\frac{(1 - \mathbf{y}_{aux1}^2)^2}{(i\omega \mathbf{y}_{aux1} + \dot{\mathbf{y}}_{aux1})^2} - f^2(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}, \dot{\mathbf{y}}_{aux1}, \dot{\mathbf{y}}_{aux2})} \right), \end{aligned}$$

$$\begin{aligned} f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}, \dot{\mathbf{y}}_{aux1}, \dot{\mathbf{y}}_{aux2}) &= \frac{v^2}{8d_{A_{12}}} \left(\frac{(1 - \mathbf{y}_{aux1}^2)^2}{(i\omega \mathbf{y}_{aux1} + \dot{\mathbf{y}}_{aux1})^2} - \frac{(1 - \mathbf{y}_{aux2}^2)^2}{(i\omega \mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2})^2} \right) \\ &+ \frac{d_{A_{12}}}{2}. \end{aligned} \quad (3.108)$$

Cette équation permet en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées d'obtenir l'estimateur de la position du robot.

De plus nous avons :

$$\Theta = \frac{z_{A_1} - z}{|z_{A_1} - z| \times \mathbf{y}_{aux1}} \quad (3.109)$$

Cette équation permet en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées d'obtenir l'estimateur de l'orientation du robot \square

3.3.7 Deux points d'intérêt (mesures : une élévation et un azimut)

Dans ce cas d'étude, HM5, le robot observe deux points d'intérêt A_1 et A_2 et mesure l'élévation relative par rapport à l'un (β_1 ou β_2) et l'azimut relatif par rapport au deuxième (α_1 ou α_2). Le modèle de mesure est le suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Theta}, \quad (3.110)$$

$$\mathbf{y}_{m2} = \frac{h_{A_2}}{|z_{r,2}|}. \quad (3.111)$$

$$(3.112)$$

3.3.7.1 Étude de la localisabilité

Théorème 3.12 *Un robot de type unicycle qui mesure l'azimut entre lui-même et un point d'intérêt et l'élévation entre lui-même et un autre point d'intérêt est localisable. Il peut donc reconstruire sa posture mais cette reconstruction dépend du vecteur d'entrée qui ne peut donc pas être reconstruit.*

Démonstration. Cherchons une relation algébrique de la forme (3.15). Pour cela, construisons à partir des mesures les deux sorties auxiliaires suivantes, à partir de HM5, quand $h_{A_\ell} \neq 0, \ell = 1, 2$:

$$\mathbf{y}_{aux1} = |z_{r,1}|, \quad (3.113)$$

$$\mathbf{y}_{aux2} = \frac{z_{r,2}}{\Theta |z_{r,2}|}. \quad (3.114)$$

Maintenant en réutilisant les résultats donnés dans le cas HM4 (voir l'équation (3.99)), nous avons :

$$|z_{r,2}| = \frac{-v(1 - \mathbf{y}_{aux2}^2)}{2(i\omega \mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2})}. \quad (3.115)$$

Prenons \mathcal{C}_ℓ le cercle centré en z_{A_ℓ} de rayon $|z_{r,2}|$ donné par l'équation (3.115) et \mathbf{y}_{aux1} . Nous avons $\mathcal{C}_{aux1} \cap \mathcal{C}_{aux2} \neq \emptyset$, dès que :

1. C1 : $|z_{A_1} - z_{A_2}| < |z_{r,1}| + |z_{r,2}|$,
2. C2 : $|z_{A_1} - z_{A_2}| > ||z_{r,1}| - |z_{r,2}||$.

Comme les deux $|z_{r,\ell}|$ ne peuvent pas être simultanément nuls, il faut distinguer les deux cas suivants :

1. $|z_{r,\ell}| \neq 0, \ell = 1, 2$: alors les conditions C1-C2 sont satisfaites. Clairement, nous avons $|z_{r,\ell}| \neq 0, \ell = 1, 2$ si et seulement si $v \neq 0$ et $\mathbf{y}_{aux2} \neq 1$. Si toutes ces conditions sont satisfaites alors, $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$ et z est l'un des deux points d'intersection des deux cercles \mathcal{C}_1 et \mathcal{C}_2 ,
2. Si seulement l'un des $|z_{r,\ell}|$ est nul alors le robot est situé en A_ℓ avec ℓ tel que $z_{r,\ell} = 0$.

Il ne reste plus qu'à estimer Θ (et donc θ) qui peut être obtenu directement en utilisant (3.114) (dès que $v \neq 0$ et $\mathbf{y}_{aux2} \neq 1$).

Nous avons donc trouvé une relation algébrique de la forme (3.15). Le résultat est alors direct par application du théorème 3.2

Il n'est pas possible dans ce cas d'obtenir les vitesses linéaire et angulaire du robot, v, ω à partir des mesures. En effet, le rayon du cercle dépend des mesures mais aussi de la vitesse linéaire et de la vitesse angulaire. Le calcul des dérivées

première et seconde de z nécessaires au calcul de v et ω (voir les équations (3.5) et (3.6)) amène à une relation de la forme :

$$\dot{z} = g_z(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}) \quad (3.116)$$

$$v = g_v(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}) \quad (3.117)$$

$$\ddot{z} = g_{\ddot{z}}(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}, \ddot{v}, \ddot{\omega}, \ddot{\mathbf{y}}_{fl}) \quad (3.118)$$

$$\omega = g_\omega(v, \omega, \mathbf{y}_{fl}, \dot{v}, \dot{\omega}, \dot{\mathbf{y}}_{fl}, \ddot{v}, \ddot{\omega}, \ddot{\mathbf{y}}_{fl}) \quad (3.119)$$

dans laquelle les variables inconnues apparaissent. Il n'est donc pas possible pour les vitesses de trouver une relation de la forme (3.15) et il n'est donc pas possible d'estimer les vitesses sous l'hypothèse de mesure HM5. \square

3.3.7.2 Estimateur d'état

Théorème 3.13 *Un robot de type unicycle qui mesure l'élévation relative entre lui-même et un point d'intérêt et l'azimut relatif entre lui-même et un autre point d'intérêt, (3.110)-(3.111), peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned} \hat{z} &= z_{A_1} + z_{A_{12}} \left(f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\dot{\mathbf{y}}}_{aux2}) \right) \\ &\pm z_{A_{12}} i \sqrt{\hat{\mathbf{y}}_{aux1}^2 - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\dot{\mathbf{y}}}_{aux2})}, \end{aligned} \quad (3.120)$$

$$\hat{\Theta} = \frac{z_{A_2} - z}{\hat{\mathbf{y}}_2(|z_{A_2} - z|)} \quad (3.121)$$

$$\begin{aligned} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\dot{\mathbf{y}}}_{aux2}) &= \frac{\hat{\mathbf{y}}_{aux1}^2}{2d_{A_{12}}} - \frac{v^2(1 - \hat{\mathbf{y}}_{aux2}^2)^2}{8d_{A_{12}}(i\omega\hat{\mathbf{y}}_{aux2} + \hat{\dot{\mathbf{y}}}_{aux2})^2} \\ &+ \frac{d_{A_{12}}}{2} \end{aligned} \quad (3.122)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (3.123)$$

Démonstration. À partir de l'hypothèse de mesure HM5, quand $z_{A_{aux1}} \neq 0$ et $v \neq 0$ il est possible de prendre les sorties auxiliaires suivantes (3.113)-(3.114) :

$$\begin{aligned} \mathbf{y}_{aux1} &= |z_{r,1}|, \\ \mathbf{y}_{aux2} &= \frac{z_{r,2}}{\Theta |z_{r,2}|}. \end{aligned}$$

En utilisant l'équation (A.16) donnée en annexe A.4, avec les rayons des cercles suivants $r_{aux1} = \mathbf{y}_{aux1}$ et $r_{aux2} = \frac{-v(1 - \mathbf{y}_{aux2}^2)}{2(i\omega\mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2})}$, nous obtenons :

$$\begin{aligned} z &= z_{A_1} + z_{A_{12}} (f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}, \dot{\mathbf{y}}_{aux2})) \\ &\pm z_{A_{12}} \left(i \sqrt{\mathbf{y}_{aux1}^2 - f^2(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}, \dot{\mathbf{y}}_{aux2})} \right). \end{aligned} \quad (3.124)$$

$$\begin{aligned} f(\mathbf{y}_{aux1}, \mathbf{y}_{aux2}, \dot{\mathbf{y}}_{aux2}) &= \frac{\mathbf{y}_{aux1}^2}{2d_{A_{12}}} - \frac{v^2(1 - \mathbf{y}_{aux2}^2)^2}{8d_{A_{12}}(i\omega\mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2})^2} \\ &+ \frac{d_{A_{12}}}{2} \end{aligned} \quad (3.125)$$

L'équation (3.124) permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de la position du robot. Nous pouvons alors donner l'expression de Θ :

$$\Theta = \frac{z_{A_2} - z}{y_2(|z_{A_2} - z|)} \quad (3.126)$$

Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de l'orientation du robot. \square

3.3.8 Synthèse des résultats obtenus

La table 3.2 synthétise les résultats obtenus pour la localisabilité d'un robot de type unicycle observant un ou deux points d'intérêt. Celle-ci résume aussi les équations des estimateurs d'état et de commande présentées dans la section correspondante. Rappelons que les mesures sont entachées d'un bruit additif. Ainsi, dans chacune des expressions algèbro-différentielles présentées, il est nécessaire d'ajouter ce bruit. Cependant, le différenciateur en environnement bruité permet de filtrer ce bruit. C'est pourquoi nous marquons d'un chapeau toute variable estimée à partir des mesures bruitées après le passage dans un dérivateur numérique d'ordre adapté.

TABLE 3.2 – *Tableau récapitulatif mesures/ localisabilité/ estimateur*

Cas	Nombre de points d'intérêt	Mesures	Conditions	\hat{z}	$\hat{\Theta}$	\hat{v}	$\hat{\omega}$
<u>HM1</u>	1	(α_1, β_1) \Leftrightarrow (y_{m1}, y_{m2}) (3.17)-(3.18)	$h_{A_1} \neq 0$	X	X	(3.26)	(3.27)
<u>HM1 BIS</u>	1 + cible	$(\alpha_1, \beta_1, \alpha_2, \beta_2)$ \Leftrightarrow $(y_{m1}, y_{m2}, y_{m3}, y_{m4})$ (3.32)-(3.35) avec h_{A_2} et z_{A_2} inconnu.	$h_{A_\ell} \neq 0,$ $\ell = 1, 2$	(3.39)	(3.38)	(3.37)	(3.40)
<u>HM2</u>	1 + Boussole	$(\alpha_1, \beta_1, \theta)$ \Leftrightarrow (y_{m1}, y_{m2}, y_{m3}) (3.54)-(3.56)	$h_{A_1} \neq 0$	(3.61)	(3.62)	(3.63)	(3.64)
<u>HM3</u>	2	(β_1, β_2) \Leftrightarrow (y_1, y_2) (3.70)-(3.71)	$h_{A_\ell} \neq 0,$ $\ell = 1, 2,$ $v \neq 0$	(4.95)	(4.96)	(4.97)	(4.98)
<u>HM4</u>	2	(α_1, α_2) \Leftrightarrow (y_{m1}, y_{m2}) (3.91)-(3.92)	$v \neq 0$ $y_{m1} \neq 1$ et $y_{m2} \neq 1$	(4.170)	(4.171)	X	X
<u>HM5</u>	2	(α_1, β_2) \Leftrightarrow (y_{m1}, y_{m2}) (3.110)-(3.111)	$h_{A_1} \neq 0$ $v \neq 0$ $y_{m2} \neq 1$	(3.120)	(3.121)	X	X

3.4 SIMULATION ET EXPÉRIMENTATION

Les différents estimateurs de posture et éventuellement de commande présentés dans la section 3.3 ont été testés sur des scénarios simulés et comparés au filtre de Kalman Étendu. Tout d'abord, les paramètres des différentes simulations, les notations utilisées et les différences d'implémentations en fonction des cas d'étude seront présentés. Puis une étude numérique de chacun des algorithmes sera donnée. Nous comparerons dans le cadre de notre simulation les résultats obtenus par les algorithmes présentés dans ce travail avec les résultats obtenus par un filtre de Kalman Étendu présenté dans l'introduction. Finalement, nous prendrons le meilleur des algorithmes proposés pour effectuer une expérimentation sur un robot réel équipé d'une caméra (le meilleur signifiant le plus facile à implémenter et/ou donnant les meilleurs résultats numériques en simulation en terme d'erreur ou de temps de calcul).

3.4.1 Paramètres de simulation et cas d'étude

Les expérimentations sont menées dans chacun des cas étudiés dans la section 3.3. Le lecteur pourra se reporter à la table 3.1 dans laquelle sont rappelées les différentes hypothèses de mesures. Les expérimentations sont menées sur Matlab sur une machine virtuelle présentant 1024 Mo de mémoire vive et un processeur. Pour chacun des cas d'étude, nous appelons l'implémentation de l'estimateur correspondant NON-A-HM i , $i = 1, 1BIS, 2, 3, 4, 5$ pour les implémentations basées sur l'utilisation des estimateurs rappelés dans la table 3.2 en lien avec le dérivateur numérique présenté dans la section 2.2.4 et EKF-HM i , $i = 2, 3, 4, 5$ pour les implémentations basées sur l'EKF. Remarquons que nous ne présentons pas de résultats pour l'EKF dans le cas EKF-HM1 car l'EKF n'est pas adapté pour estimer les commandes envoyées au robot dans le cas où elles ne sont pas connues ni dans le cas EKF-HM1-BIS car l'EKF n'est pas adapté pour l'ajout d'équation permettant de lever le défaut de localisabilité.

Dans le cas EKF-HM2 et EKF-HM3 où les commandes envoyées ne sont pas nécessaires pour notre algorithme (NONA-HM2 et NONA-HM3), l'EKF est implémenté sans l'étape de prédiction, seule l'étape de mise à jour est effectuée. Dans le cas EKF-HM4 et EKF-HM5 où les commandes sont nécessaires pour nos algorithmes (NON-A-HM4 et NON-A-HM5) les commandes sont supposées connues et l'EKF est implémenté normalement. Pour que les deux types de méthodes puissent être comparables, l'EKF est initialisé à $10m$ de la vraie position et la matrice de covariance de l'initialisation est remplie avec des grandes valeurs sur la diagonale (en effet nos algorithmes ne nécessitent pas d'être initialisés).

Chacune des simulations est répétée 200 fois (chaque simulation produisant un jeu de données) de manière à pouvoir calculer les paramètres statistiques des erreurs obtenues (notamment moyenne et variance de l'erreur) en suivant la loi des grands nombres. La période d'échantillonnage des mesures est de $T_s = 1/30sec$. Le bruit de mesure est réparti de façon uniforme sur l'intervalle $[-0.0017 \ 0.0017][rad]$, qui est un intervalle d'erreur classique pour une caméra monoculaire (0.1 degrés d'erreur). Pour chaque simulation, les points d'intérêt sont placés de façon aléatoire dans un cube de l'espace préalablement défini.

L'algorithme de dérivation nécessite quelques paramètres de réglages, notamment la longueur de la fenêtre d'intégration que nous fixons à 80 échantillons, les paramètres $(\kappa, \mu) = (0, 0)$ (de manière à avoir des retards d'estimations identiques quel que soit l'ordre de dérivation) et l'ordre de troncature de la série de Taylor que nous fixons à l'ordre de dérivation souhaité ($N = n$) (pour augmenter la robustesse par rapport au bruit par rapport à $N > n$).

Nous étudions deux scénarios de simulation. Dans le premier scénario, nous supposons avoir une bonne connaissance des propriétés statistiques du bruit, nous donnons donc à l'EKF une matrice de covariance du bruit correspondant au bruit appliqué. Dans le deuxième scénario, nous supposons ne pas avoir une bonne connaissance du bruit ; nous remplissons donc la matrice de covariance du bruit avec de grandes valeurs sur la diagonale pour indiquer cette mauvaise connaissance à l'algorithme.

3.4.2 Résultats numériques

La table 3.3 donnent les résultats numériques obtenus pour chaque algorithme dans les deux simulations. Pour la première simulation, où nous supposons avoir une bonne connaissance statistique du bruit, l'EKF donne des résultats meilleurs

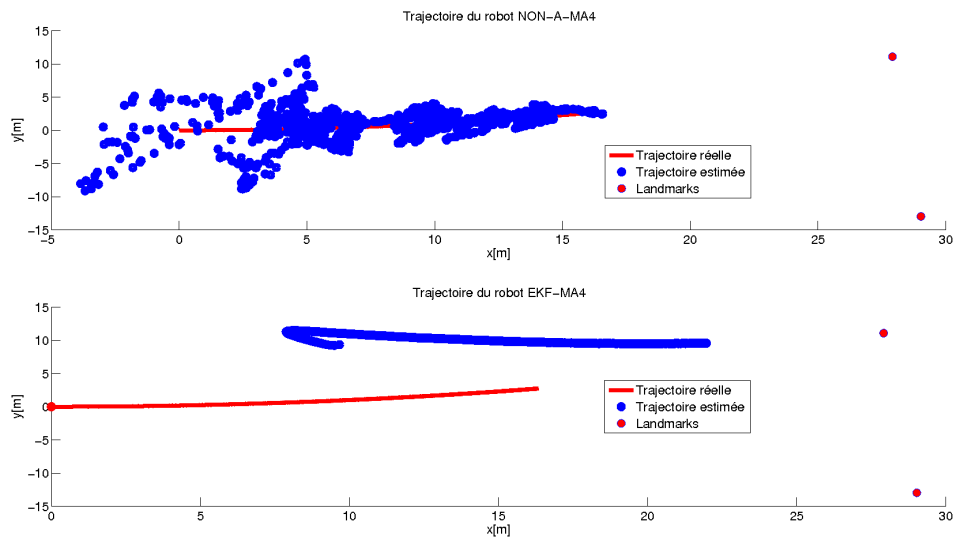


FIGURE 3.2 – Comparaison des algorithmes NON-A-HM4 et EKF-HM4

dans le cas HM2 mais moins bons dans le cas HM3, HM4 et HM5. L'algorithme NON-A-HM1BIS donne des résultats comparables avec le cas HM3 alors même que la difficulté est plus grande (présence d'un défaut de localisabilité levé grâce à l'introduction d'un point d'intérêt de coordonnées inconnues). On voit que dans les cas HM4, HM5 les résultats numériques sont mauvais pour les deux algorithmes, cependant, notre algorithme donne des résultats centrés autour de la trajectoire réelle alors même que l'EKF n'arrive pas à converger vers la trajectoire réelle (il n'arrive pas à récupérer l'erreur d'initialisation). Cet effet est présenté sur la figure 3.2 où au dessus se trouvent les résultats donnés par l'algorithme NON-A-HM4 et en dessous celui donné par l'algorithme EKF-HM4 pour un jeu de données.

Pour la deuxième simulation où nous supposons ne pas avoir une bonne connaissance statistique du bruit (grandes valeurs sur la diagonale de la matrice de covariance du bruit), nos algorithmes donnent des résultats systématiquement meilleurs que ceux donnés par l'EKF. Ces résultats sont identiques à ceux donnés dans la simulation précédente, nos algorithmes n'ont pas besoin de connaître les propriétés statistiques du bruit pour fonctionner.

En termes de temps de calcul nos algorithmes sont meilleurs dans les cas NON-A-HM2 et NON-A-HM1BIS et moins bons dans les autres cas. En effet, la fonction calculant l'intersection de deux cercles présente un temps de calcul élevé.

3.4.3 Résultats expérimentaux

Nous avons implémenté l'algorithme NON-A-HM1BIS sur un robot réel. En effet cet algorithme est simple à utiliser (un seul point d'intérêt de coordonnées connues et un point d'intérêt de coordonnées inconnues), ne nécessite pas de connaître les commandes envoyées au robot et ne demande pas à connaître l'orientation du robot (contrairement au cas HM2). L'expérimentation a été menée sur un robot conçu à l'ISEN (voir la figure de gauche de 3.3). C'est un robot de type unicycle équipé d'une caméra Imaging Source préalablement calibrée de manière à connaître précisément sa distance focale et la position du centre de l'image. Une des images acquises par la caméra est montrée sur la figure de droite de 3.3. Les

TABLE 3.3 – Résultats numériques

(a) Simulation 1

Algorithmes	Résultats				
	erreur moyenne en distance [m]	variance moyenne de l'erreur en distance [m ²]	erreur moyenne sur v [m.s ⁻¹]	erreur moyenne sur ω [rad.s ⁻¹]	Temps d'exécution moyen [s]
<u>NON-A-HM1BIS</u>	0.2117	0.0035	0.0334	0.0015	0.1706
<u>NON-A-HM2</u>	0.1421	0.0033	0.0287	3.2435e-006	0.0747
<u>EKF-HM2</u>	0.1086	0.0157	X	X	0.2278
<u>NON-A-HM3</u>	0.1853	0.0033	0.0278	0.2485	0.6236
<u>EKF-HM3</u>	0.3455	0.5219	X	X	0.1477
<u>NON-A-HM4</u>	1.7140	1.9848	X	X	0.6236
<u>EKF-HM4</u>	10.8567	2.5690	X	X	0.2300
<u>NON-A-HM5</u>	1.3338	1.5110	X	X	0.6299
<u>EKF-HM5</u>	7.0835	0.3011	X	X	0.2322

(b) Simulation 2

Algorithmes	Résultats				
	erreur moyenne en distance [m]	variance moyenne de l'erreur en distance [m ²]	erreur moyenne sur v [m.s ⁻¹]	erreur moyenne sur ω [rad.s ⁻¹]	Temps d'exécution moyen [s]
<u>NON-A-HM1BIS</u>	0.2124	0.0033	0.329	0.0015	0.1785
<u>NON-A-HM2</u>	0.1397	0.0031	0.0284	3.2435e-006	0.0743
<u>EKF-HM2</u>	0.4319	1.0527	X	X	0.2399
<u>NON-A-HM3</u>	0.1794	0.0031	0.0270	0.2433	0.6605
<u>EKF-HM3</u>	4.5987	10.2200	X	X	0.1525
<u>NON-A-HM4</u>	1.7433	2.0133	X	X	0.6569
<u>EKF-HM4</u>	10.8525	2.3314	X	X	0.2387
<u>NON-A-HM5</u>	1.3609	1.4703	X	X	0.6567
<u>EKF-HM5</u>	6.4227	2.0095	X	X	0.2396

points d'intérêt sont représentés par un point rouge de l'environnement. Seul le point de droite est de coordonnées connues. Ces points sont extraits du flux vidéo au fur et à mesure de l'expérience. Cette expérience a été répétée sur huit jeux de données sur la même trajectoire (voir la figure 3.4). Les résultats sont compilés dans la table 3.4. Nous obtenons des résultats concordant avec les simulations, cependant certaines erreurs pourraient être levées si nous arrivions à obtenir un bon calibrage de la caméra, ce qui est loin d'être évident.

TABLE 3.4 – Résultats expérimentaux de l'algorithme NON-A

	erreur moyenne en distance [cm]	variance moyenne de l'erreur en distance [cm ²]
NON-A- HM1-BIS	11	25

3.5 FUSION DANS LE CAS MULTI-POINTS D'INTÉRÊT

Les travaux de cette section ont fait l'objet d'une publication à IROS 2012 [136]. L'objectif de cette partie est de proposer un algorithme de fusion de données dans le cas où plusieurs points d'intérêt sont disponibles. Nous allons effectuer cette étude

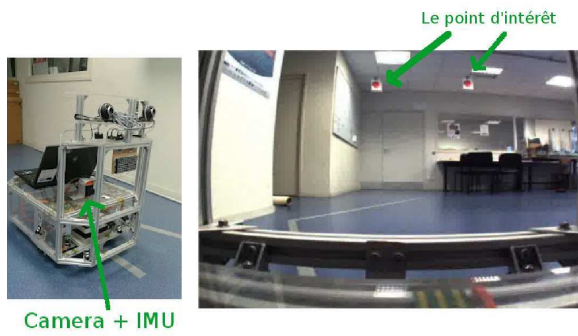


FIGURE 3.3 – *Le robot - Une image de la caméra*

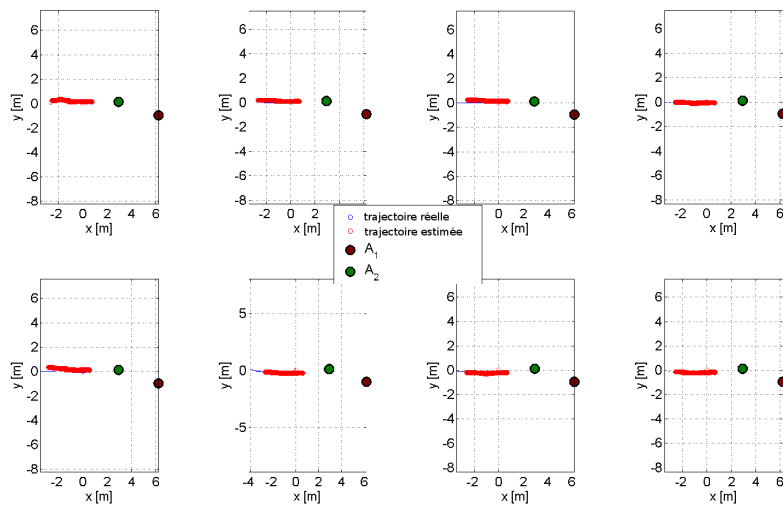


FIGURE 3.4 – *Trajectoire utilisée pour les expérimentations*

dans le cas HM2. L'avantage de ce modèle de mesure est qu'il permet de prouver la faisabilité de l'approche sans risquer de biaiser les résultats par un mauvais choix d'association de points d'intérêts (association qu'il faut faire dans tous les autres cas qui nécessite au moins deux points d'intérêts pour fonctionner).

3.5.1 Hypothèse de travail

Dans le cas HM2 nous supposons que le robot mesure les angles relatifs entre lui et un point d'intérêt ainsi que son orientation à l'aide par exemple d'une boussole. Rappelons ici le modèle de mesure correspondant :

$$\mathbf{y}_{m1\ell} = \frac{z_{r,\ell}}{|z_{r,\ell}|\Theta}, \quad (3.127)$$

$$\mathbf{y}_{m2\ell} = \frac{h_{A_\ell}}{|z_{r,\ell}|}, \quad (3.128)$$

$$\mathbf{y}_{m3} = \Theta. \quad (3.129)$$

Nous ajoutons l'indice ℓ de manière à étendre l'estimateur proposé pour un point d'intérêt au cas multi-points d'intérêt, chacune des mesures est associée au point d'intérêt correspondant A_ℓ . Nous remarquerons que \mathbf{y}_{m3} n'est pas affectée par un indice car nous supposons que la mesure de l'orientation est indépendante du point d'intérêt observé et peut être mesurée en utilisant par exemple une boussole.

Nous supposons que l'erreur sur chacune des mesures est bornée et nous notons cette borne $\Delta\eta_{\alpha\ell}$ pour l'erreur sur $\mathbf{y}_{m1\ell}$, $\Delta\eta_{\beta\ell}$ pour l'erreur sur $\mathbf{y}_{m2\ell}$ et $\Delta\eta_\Theta$ pour l'erreur sur la mesure de l'orientation. Ainsi nous avons :

$$\exp(i\alpha_{m,\ell}) \in [\exp(i\alpha_\ell) - \Delta\eta_{\alpha\ell} ; \exp(i\alpha_\ell) + \Delta\eta_{\alpha\ell}]$$

3.5.2 Étude de sensibilité de la solution proposée

Les sorties auxiliaires utilisées pour l'étude de ce cas sont les suivantes :

$$\begin{aligned} \mathbf{y}_{1\ell} &= \frac{\mathbf{y}_{m1\ell}\mathbf{y}_{m3}h_{A_\ell}}{\mathbf{y}_{m2\ell}} = z_{r,\ell}, \\ \mathbf{y}_2 &= \frac{\mathbf{y}_{m3}}{\mathbf{y}_{m3}} = \Theta. \end{aligned} \quad (3.130)$$

Étendons maintenant au cas multi-points d'intérêt le résultat obtenu dans le cadre mono-point d'intérêt.

Un robot de type unicycle mesurant son orientation et les angles relatifs entre un point d'intérêt A_ℓ et lui-même est localisable. Un estimateur d'état et de commandes est alors donné par les équations suivantes :

$$z = z_{A_1} - \mathbf{y}_{1\ell}, \quad (3.131)$$

$$\Theta = \mathbf{y}_2, \quad (3.132)$$

$$v = -\frac{\dot{\mathbf{y}}_{1\ell}}{\mathbf{y}_2}, \quad (3.133)$$

$$\omega = \frac{-i\dot{\mathbf{y}}_2}{\mathbf{y}_2}. \quad (3.134)$$

La sensibilité de l'estimation de $z_{r,\ell}$ par rapport aux erreurs de mesure est donnée par

$$S_{z_{r,\ell}} = \frac{\mathbf{y}_{m3\ell}h_{A_\ell}}{\mathbf{y}_{m2\ell}}\Delta\eta_{\alpha\ell} + \frac{\mathbf{y}_{m1\ell}h_{A_\ell}}{\mathbf{y}_{m2\ell}}\Delta\eta_\Theta - \frac{\mathbf{y}_{m1\ell}\mathbf{y}_{m3\ell}h_{A_\ell}}{\mathbf{y}_{m2\ell}^2}\Delta\eta_{\beta\ell} \quad (3.135)$$

De même la sensibilité de l'estimation de Θ par rapport aux erreur de mesures est donnée par

$$S_{\Theta} = \Delta\eta_{y_2} = \Delta\eta_{\Theta} \quad (3.136)$$

Étudions un peu plus en détail la sensibilité sur l'estimation de z . En utilisant l'expression des sorties auxiliaires et le modèle de mesure :

$$S_{z_{r,\ell}} = \frac{z_{r,\ell}}{\mathbf{y}_{m1}} \Delta\eta_{\alpha\ell} + \frac{z_{r,\ell}}{\mathbf{y}_{m3}} \Delta\eta_{\Theta} - \frac{z_{r,\ell}}{\mathbf{y}_{m2}} \Delta\eta_{\beta\ell} \quad (3.137)$$

$$S_{z_{r,\ell}} = |z_{r,\ell}| \Theta \Delta\eta_{\alpha\ell} + \frac{z_{r,\ell}}{\Theta} \Delta\eta_{\Theta} - \frac{|z_{r,\ell}| z_{r,\ell}}{h_{A_\ell}} \Delta\eta_{\beta\ell} \quad (3.138)$$

La position du point d'intérêt ne joue pas sur l'orientation du robot, on peut donc s'affranchir dans notre étude de la variable Θ . On voit alors que plus le point d'intérêt est proche du robot plus $|z_{r,\ell}|$ est petit et plus la sensibilité est faible. De même, plus le point d'intérêt est haut plus la sensibilité de l'estimation diminue. Introduisons maintenant la sensibilité cumulative globale que nous utiliserons pour proposer un algorithme de fusion de données ;

$$|S_{z_{r,\ell}}| = \left(||z_{r,\ell}| \Theta| + ||z_{r,\ell}| \Theta^{-1}| + \left| \frac{|z_{r,\ell}| z_{r,\ell}}{h_{A_\ell}} \right| \right) \quad (3.139)$$

3.5.3 Algorithme de fusion dans le cas de plusieurs points d'intérêt

La première possibilité pour fusionner les résultats dans le cas où plusieurs points d'intérêt donnent une estimation simultanée de la position du robot consiste à faire une moyenne des résultats obtenus. Cependant, l'étude menée précédemment sur la sensibilité de l'estimation en fonction de la position des point d'intérêt permet d'envisager une autre possibilité. La figure 3.5 montre l'algorithme envisagé. Pour chaque point d'intérêt A_ℓ et à chaque instant, une estimation de la position relative robot-point d'intérêt $\hat{z}_{r,\ell}$ et l'orientation du robot $\hat{\Theta}$ est calculée. Cette position relative permet de calculer :

$$|S_{\hat{z}_{r,\ell}}| = \left| |\hat{z}_{r,\ell}| \hat{\Theta} \right| \Delta\eta_{\alpha\ell} + \left| \frac{\hat{z}_{r,\ell}}{\hat{\Theta}} \right| \Delta\eta_{\Theta} + \left| \frac{|\hat{z}_{r,\ell}| \hat{z}_{r,\ell}}{h_{A_\ell}} \right| \Delta\eta_{\beta\ell}. \quad (3.140)$$

Chacune des estimations de la position relative robot-point d'intérêt $\hat{z}_{r,\ell}$ nous permet d'avoir une estimation de la position du robot notée \hat{z}_ℓ . C'est une fusion de l'ensemble des estimations que nous faisons pour obtenir une estimation globale de la position du robot :

$$\hat{z} = \frac{\sum_{\ell=1}^n \hat{z}_\ell \times \frac{1}{S_{\hat{z}_{r,\ell}}}}{\sum_{\ell=1}^n \frac{1}{S_{\hat{z}_{r,\ell}}}}. \quad (3.141)$$

3.5.4 Résultats en simulation

3.5.4.1 Simulation avec un bloc de points d'intérêt

Nous utilisons les mêmes paramètres de simulation que précédemment. La seule différence réside dans le nombre de points d'intérêt, qui augmente en fonction des simulations pour voir l'impact de l'algorithme de fusion de données. Nous

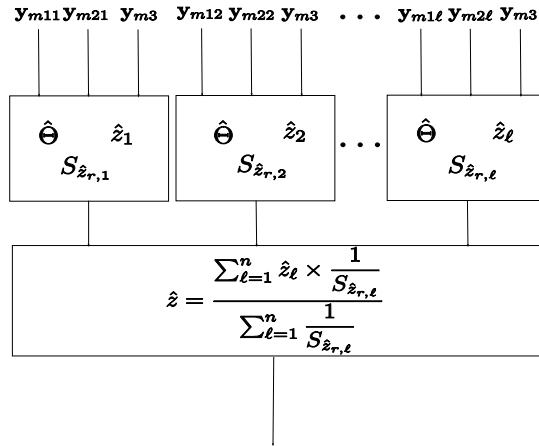


FIGURE 3.5 – Localisation multi-points d'intérêt par fusion

appelons NON-A-MOYENNE l'algorithme qui effectue une moyenne des estimations dans le cas multi-points d'intérêt et NON-A-SENSIBILITE l'algorithme présenté dans la section précédente. Nous comparons encore les résultats avec l'EKF. La figure 3.6 montre la trajectoire utilisée pour un jeu de données. La position de chaque point d'intérêt est choisie de façon aléatoire dans un bloc (en vert sur la figure), et ce pour chaque simulation. Dans le premier cas, nous utilisons un seul bloc de points d'intérêt. La table 3.5 donne les résultats numériques.

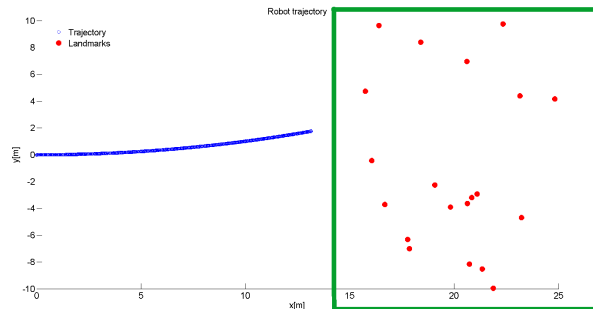


FIGURE 3.6 – Simulation avec un bloc de points d'intérêt

nombre de points d'intérêt	1	5	10	50	100
$e_{NON-A-MOYENNE}(cm)$	1.6606	1.2131	1.0814	0.9422	0.9342
$e_{NON-A-SENSIBILITE}(cm)$	1.6032	1.1634	1.0446	0.9363	0.9332
$e_{EKF}(cm)$	2.3552	1.8137	1.7197	1.3939	1.2210
$\sigma_{NON-A-MOYENNE}(cm^2)$	0.1331	0.0360	0.0237	0.0203	0.0152
$\sigma_{NON-A-SENSIBILITE}(cm^2)$	0.1215	0.0297	0.0204	0.0185	0.0140
$\sigma_{EKF}(cm^2)$	0.1510	0.0161	0.0121	0.0038	0.0018
$t_{NON-A-MOYENNE}$	0.2706	0.4255	0.4721	0.4354	0.3863
t_{EKF}					
$t_{NON-A-SENSIBILITE}$	0.2328	0.4010	0.4200	0.4032	0.3592
t_{EKF}					

TABLE 3.5 – Résultats comparatifs des trois algorithmes avec un bloc de points d'intérêt

Les algorithmes NON-A donnent des résultats équivalents ou meilleurs que l'EKF. Le principal point d'amélioration réside dans le temps de calcul qui est significativement réduit.

3.5.4.2 Simulation avec deux blocs de points d'intérêt

La figure 3.7 montre la trajectoire utilisée pour les simulations. La position de chacun des points d'intérêts est choisie aléatoirement parmi des deux blocs possibles (en vert sur la figure). Le deuxième bloc, sur la partie droite de la figure, est fort éloigné du robot et donnera donc de moins bons résultats que le premier (sur la partie gauche de la figure) La table 3.6 donne les résultats numériques.

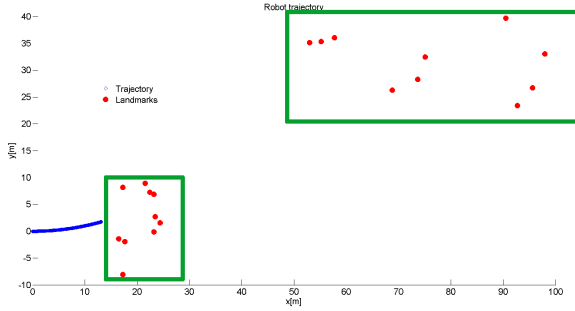


FIGURE 3.7 – Simulation avec deux blocs de points d'intérêt

nombre de points d'intérêt	2	6	10	50	100
$e_{NON-A-MOYENNE}(cm)$	15.2570	9.9732	7.6324	4.1055	3.3923
$e_{NON-A-SENSIBILITE}(cm)$	2.9550	1.8344	1.5591	1.1697	1.1406
$e_{EKF}(cm)$	3.2767	1.9699	1.7670	1.3647	1.2168
$\sigma_{NON-A-MOYENNE}(cm^2)$	60.2199	9.9514	3.4392	0.5371	0.2815
$\sigma_{NON-A-SENSIBILITE}(cm^2)$	0.6881	0.1182	0.0832	0.0292	0.0285
$\sigma_{EKF}(cm^2)$	1.1092	0.0428	0.0165	0.0026	0.0016
$t_{NON-A-MOYENNE}$	0.3718	0.4081	0.4581	0.4286	0.3747
t_{EKF}					
$t_{NON-A-SENSIBILITE}$	0.3104	0.3919	0.4231	0.4103	0.3621
t_{EKF}					

TABLE 3.6 – Résultats comparatifs des trois algorithmes avec deux blocs de points d'intérêt

L'algorithme NON-A-SENSIBILITE donne de meilleurs résultats numériques que l'algorithme NON-A-MOYENNE en terme de moyenne de l'erreur et de variance de l'erreur. Qui plus est, cet algorithme donne aussi de meilleurs résultats que l'EKF. L'algorithme proposé pour la fusion de données améliore grandement les résultats obtenus par notre méthode.

CONCLUSION DU CHAPITRE

Nous avons étudié dans ce chapitre la localisation d'un robot mobile unicycle (c'est à dire un robot de type $(2, 0)$ en suivant la classification donnée dans la section 2.1.4.6). Cette étude s'effectue en deux étapes : une étape où l'on analyse la localisabilité du robot sous une hypothèse de mesures donnée, et une étape où l'on construit un estimateur d'état pour le robot à partir des mesures. En préambule à ce chapitre, la notion de localisabilité a été présentée. Ensuite, le modèle du robot utilisé a été rappelé et les cinq hypothèses des mesures du capteur utilisé ont été présentées. Il était alors possible d'effectuer une étude exhaustive des cinq cas d'étude tant du point de vue de la localisabilité que de la construction de l'estimateur d'état correspondant. Cette analyse effectuée, une étude en simulation et sur robot réel a été effectuée pour prouver le bon fonctionnement de nos algorithmes. Pour la simulation, une comparaison avec les résultats donnés par un filtre de Kalman Étendu permet de conclure quant à l'efficacité de nos algorithmes en terme de moyenne et de variance de l'erreur d'estimation et en terme de temps de calcul par

rapport à la littérature. Finalement en guise de conclusion à ce chapitre, une étude de la sensibilité de l'algorithme proposé pour l'un des cas d'étude a été faite, ces résultats permettant de proposer un algorithme de fusion de données pour le cas où plusieurs points d'intérêt sont disponibles.

LOCALISATION DANS UN CADRE GÉNÉRAL

SOMMAIRE	
4.1	RAPPEL DES MODÈLES CINÉMATIQUES 103
4.1.1	Robot de type $(2, 0)$ 103
4.1.2	Robot de type $(3, 0)$ 103
4.1.3	Robot de type $(2, 1)$ 103
4.1.4	Robot de type $(2, 1)$ 103
4.2	LOCALISATION 104
4.2.1	Un point d'intérêt et une cible (mesures : azimuts et élévations) . 104
4.2.2	Un point d'intérêt avec boussole (mesures : azimut, élévation et θ) 108
4.2.3	Deux points d'intérêt (mesures : deux élévations) 110
4.2.4	Deux points d'intérêt (mesures : deux azimuts) 115
4.2.5	Deux points d'intérêt (mesures : une élévation et un azimut) . . 118
4.3	RÉSULTATS DE SIMULATION 121
4.3.1	Paramètres de simulation et cas d'étude 121
4.3.2	Résultats numériques 122
CONCLUSION 122	

LA généralisation de l'étude menée sur les cinq hypothèses de mesures pour le robot unicycle dans le chapitre 3 à l'ensemble des autres types de robots dont la classification est donnée dans la section 2.1.4.6 est effectuée dans ce chapitre. Dans la section 2.2.3 les modèles cinématiques des robots avaient été réécrits sous une forme algébrique qui se prête bien à l'étude de la localisabilité et de la localisation. Nous avons montré que pour les 4 types de robots que nous étudions, les modèles cinématiques étaient semblables à celui d'un robot de type $(2, 0)$ moyennant un changement de variable adéquat. Il est donc possible de réutiliser les méthodes et algorithmes développés dans le cadre du robot unicycle pour les étendre à l'ensemble des autres robots. Il est intéressant de noter que pour l'ensemble de ces robots, l'observabilité partielle (c'est-à-dire : la possibilité d'estimer uniquement la posture du robot sans estimer les autres variables d'état), qui est équivalente à la localisabilité, entraîne systématiquement l'observabilité totale. Il est donc toujours possible d'écrire un estimateur des autres variables d'état en mesurant uniquement la posture. Des reconSTRUCTEURS d'état pour chacun des robots et dans chacun des cinq cas d'étude sont donc proposés dans ce chapitre. Les résultats obtenus sont

validés par une étude en simulation en suivant la même méthodologie que celle employée pour le robot unicycle.

4.1 RAPPEL DES MODÈLES CINÉMATIQUES

4.1.1 Robot de type (2, 0)

Le modèle cinématique de ce robot sous une écriture complexe est le suivant :

$$\begin{cases} \dot{z} = v\Theta \\ \dot{\Theta} = i\omega\Theta \end{cases} \quad (4.1)$$

avec $(z, \Theta) = (x + iy, \exp(i\theta))$ la posture du robot et (v, ω) les vitesses de commande (linéaire et angulaire).

4.1.2 Robot de type (3, 0)

Le modèle cinématique de ce robot sous une écriture complexe est le suivant :

$$\begin{cases} \dot{z} = v\Theta \\ \dot{\Theta} = i\omega\Theta \end{cases} \quad (4.2)$$

moyennant le changement de variable suivant :

$$v = v_1 + iv_2. \quad (4.3)$$

avec $(z, \Theta) = (x + iy, \exp(i\theta))$ la posture du robot (position et orientation) et (v_1, v_2, ω) les vitesses de commande (linéaire selon l'axe \vec{x} , linéaire selon l'axe \vec{y} et angulaire).

4.1.3 Robot de type (2, 1)

Le modèle cinématique de ce robot sous une écriture complexe est le suivant :

$$\begin{cases} \dot{z} = v_1\Gamma \\ \dot{\Gamma} = i\omega\Gamma \end{cases} \quad (4.4)$$

moyennant le changement de variable suivant :

$$\omega = \omega_1 + \omega_2, \quad (4.5)$$

$$v_1 = iv, \quad (4.6)$$

$$\gamma = \theta + \beta, \quad (4.7)$$

$$\Gamma = \exp(i\gamma). \quad (4.8)$$

avec $(z, \Theta) = (x + iy, \exp(i\theta))$ la posture du robot (position et orientation), $B = \exp(i\beta)$ où β est l'angle d'orientation de la roue orientable et (v, ω_1, ω_2) les vitesses de commande (linéaire, vitesse de rotation du robot et vitesse de rotation de la roue orientable).

4.1.4 Robot de type (2, 1)

Le modèle cinématique de ce robot sous une écriture complexe est le suivant :

$$\begin{cases} \dot{z} = v_2\Theta \\ \dot{\Theta} = iv_3\Theta \\ \dot{B} = i\omega B \end{cases} \quad (4.9)$$

moyennant le changement de variable suivant :

$$v_2 = iL\mathcal{I}(B)v, \quad (4.10)$$

$$v_3 = \mathcal{R}(B)v. \quad (4.11)$$

avec $(z, \Theta) = (x + iy, \exp(i\theta))$ la posture du robot (position et orientation), $B = \exp(i\beta)$ où β est l'angle d'orientation de la roue orientable et (v, ω) les vitesses de commande (où v correspond à une vitesse linéaire et ω à une vitesse angulaire).

4.2 LOCALISATION

4.2.1 Un point d'intérêt et une cible (mesures : azimuts et élévations)

4.2.1.1 Rappel du modèle de mesures

Le modèle de mesures est le suivant :

$$y_{m1} = y_{aux1} = \exp(i\alpha_1) = \frac{z_{r,1}}{\Theta |z_{r,1}|}, \quad (4.12)$$

$$y_{m2} = y_{aux2} = \tan(\beta_1) = \frac{h_{A_1}}{|z_{r,1}|}, \quad (4.13)$$

$$y_{m3} = y_{aux3} = \exp(i\alpha_2) = \frac{z_{r,2}}{\Theta |z_{r,2}|}, \quad (4.14)$$

$$y_{m4} = y_{aux4} = \tan(\beta_2) = \frac{h_{A_2}}{|z_{r,2}|}. \quad (4.15)$$

4.2.1.2 Localisation d'un robot de type $(2, 0)$

Nous avons montré qu'un tel robot est localisable et un estimateur d'état et de commande a été donné dans la section 3.3.3. Rappelons ici le résultat que nous avons obtenu :

Théorème 4.1 (voir le théorème 3.5) *Un robot de type $(2, 0)$ observant un point d'intérêt et une cible (point d'intérêt de coordonnées inconnues) et mesurant les angles relatifs entre lui-même et ces points (Équations (4.12)-(4.15)) peut reconstruire sa posture, l'altitude de la cible et ses commandes en utilisant les estimateurs suivants :*

$$\hat{v} = \frac{2h_{A_1}\hat{y}_{aux2}\hat{y}_{aux1}}{(1 + \hat{y}_{aux1}^2)\hat{y}_{aux2}^2}, \quad (4.16)$$

$$h_{A_2} = \frac{h_{A_1}\hat{y}_{aux2}\hat{y}_{aux1}}{(1 + \hat{y}_{aux1}^2)\hat{y}_{aux2}^2} \times \frac{(1 + \hat{y}_{aux3}^2)\hat{y}_{aux4}^2}{\hat{y}_{aux4}\hat{y}_{aux3}}, \quad (4.17)$$

$$\hat{\Theta}(t) = \Theta(t_0) \frac{h_{A_1} \frac{\hat{y}_{aux1}(t_0)}{\hat{y}_{aux2}(t_0)} - h_{A_2} \frac{\hat{y}_{aux3}(t_0)}{\hat{y}_{aux4}(t_0)}}{h_{A_1} \frac{\hat{y}_{aux1}(t)}{\hat{y}_{aux2}(t)} - h_{A_2} \frac{\hat{y}_{aux3}(t)}{\hat{y}_{aux4}(t)}}, \quad (4.18)$$

$$\hat{z}(t) = z_{A_1} - h_{A_1}\Theta(t) \frac{\hat{y}_{aux1}(t)}{\hat{y}_{aux2}(t)}, \quad (4.19)$$

$$(4.20)$$

$$\begin{aligned}\hat{\omega} &= -i \frac{num}{h_{A_1} \hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} \hat{\mathbf{y}}_{aux4}^2 - h_{A_2} \hat{\mathbf{y}}_{aux2}^2 \hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4}}, \quad (4.21) \\ num &= h_{A_1} \hat{\mathbf{y}}_{aux4}^2 \left(\hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} - \hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} \right) \\ &\quad + h_{A_2} \hat{\mathbf{y}}_{aux2}^2 \left(\hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4} - \hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4} \right) \\ \hat{\mathbf{y}}_{aux\ell} &= y_{f\ell}, \ell = 1, \dots, 4. \quad (4.22)\end{aligned}$$

4.2.1.3 Localisation d'un robot de type (3, 0)

Théorème 4.2 *Un robot de type (3, 0) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.12)-(4.15) (correspondant à la mesure des angles relatifs entre lui-même et un point d'intérêt et entre lui-même et une cible) peut reconstruire sa posture et l'altitude des points d'intérêt en utilisant les estimateurs suivants :*

$$\hat{z}_{A_1} = \frac{\hat{\mathbf{y}}_{aux2}^2 \left((v_1 - iv_2) \hat{\mathbf{y}}_{aux1} + \frac{v_1 + iv_2}{\hat{\mathbf{y}}_{aux1}} \right)}{2\hat{\mathbf{y}}_{aux2}}, \quad (4.23)$$

$$\hat{z}_{A_2} = \frac{\hat{\mathbf{y}}_{aux4}^2 \left((v_1 - iv_2) \hat{\mathbf{y}}_{aux3} + \frac{v_1 + iv_2}{\hat{\mathbf{y}}_{aux3}} \right)}{2\hat{\mathbf{y}}_{aux4}}, \quad (4.24)$$

$$\hat{\Theta}(t) = \Theta(t_0) \frac{\hat{z}_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t_0)}{\hat{\mathbf{y}}_{aux2}(t_0)} - \hat{z}_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t_0)}{\hat{\mathbf{y}}_{aux4}(t_0)}}{\hat{z}_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)} - \hat{z}_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t)}{\hat{\mathbf{y}}_{aux4}(t)}}, \quad (4.25)$$

$$\hat{z}(t) = z_{A_1} - h_{A_1} \Theta(t) \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)}, \quad (4.26)$$

$$\begin{aligned}\hat{\omega} &= -i \frac{num}{h_{A_1} \hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} \hat{\mathbf{y}}_{aux4}^2 - h_{A_2} \hat{\mathbf{y}}_{aux2}^2 \hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4}}, \quad (4.27) \\ num &= h_{A_1} \hat{\mathbf{y}}_{aux4}^2 \left(\hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} - \hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} \right) \\ &\quad + h_{A_2} \hat{\mathbf{y}}_{aux2}^2 \left(\hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4} - \hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4} \right) \\ \hat{\mathbf{y}}_{aux\ell} &= y_{f\ell}, \ell = 1, \dots, 4. \quad (4.28)\end{aligned}$$

Démonstration. Commençons par estimer l'altitude des points d'intérêt h_{A_1} et h_{A_2} (nous effectuons le raisonnement sur h_{A_1} il est similaire pour h_{A_2}). Calculons la dérivée temporelle de \mathbf{y}_{aux2} :

$$\dot{\mathbf{y}}_{aux2} = -h_{A_1} \frac{d|z_{r,1}|}{|z_{r,1}|^2}, \quad (4.29)$$

La dérivée de $|z_{r,\ell}|$ est :

$$\frac{d|z_{r,\ell}|}{dt} = \frac{z_{r,\ell} \bar{\dot{z}}_{r,\ell} + \bar{z}_{r,\ell} \dot{z}_{r,\ell}}{2|z_{r,\ell}|}, \quad (4.30)$$

En remplaçant dans (4.29), l'expression de la dérivée temporelle de $|z_{r,\ell}|$ avec

$\ell = 1$ et en utilisant le modèle du robot, nous obtenons :

$$\dot{\mathbf{y}}_{aux2} = \frac{h_{A_1}}{2|z_{r,1}|} \times ((v_1 - iv_2) \Theta^{-1} z_{r,1} + \overline{z_{r,1}} (v_1 + iv_2) \Theta) \quad (4.31)$$

$$\dot{\mathbf{y}}_{aux2} = \frac{\mathbf{y}_{aux2}^2}{2h_{A_1}} \times \left((v_1 - iv_2) \mathbf{y}_{aux1} + \frac{(v_1 + iv_2)}{\mathbf{y}_{aux1}} \right) \quad (4.32)$$

$$h_{A_1} = \frac{\mathbf{y}_{aux2}^2 \left((v_1 - iv_2) \mathbf{y}_{aux1} + \frac{v_1 + iv_2}{\mathbf{y}_{aux1}} \right)}{2\dot{\mathbf{y}}_{aux2}} \quad (4.33)$$

Cette équation permet, en remplaçant la sortie auxiliaire et ses dérivées par leurs valeurs numériques filtrées, d'obtenir l'estimateur de l'altitude du point d'intérêt. La suite de la démonstration est équivalente à celle donnée pour le théorème 3.5 où l'on ne garde que l'estimation de la posture et de la vitesse angulaire avec $v = v_1 + iv_2$ (il n'est pas possible d'estimer la vitesse linéaire du robot, car elle est nécessaire pour l'estimation de l'altitude des points d'intérêt). \square

4.2.1.4 Localisation d'un robot de type (2, 1)

Le modèle de mesure est décrit par les équations (4.34)-(4.37) (nous supposons que le robot est capable de mesurer l'orientation de sa roue orientable, à l'aide par exemple d'une codeuse montée sur l'axe de rotation verticale de la roue).

$$\mathbf{y}_{m1} = \mathbf{y}_1 = \exp(i\alpha_1) = \frac{z_{r,1}}{\Gamma|z_{r,1}|}, \quad (4.34)$$

$$\mathbf{y}_{m2} = \mathbf{y}_2 = \tan(\beta_1) = \frac{h_{A_1}}{|z_{r,1}|}. \quad (4.35)$$

$$\mathbf{y}_{m3} = \mathbf{y}_3 = \exp(i\alpha_2) = \frac{z_{r,2}}{\Gamma|z_{r,2}|}, \quad (4.36)$$

$$\mathbf{y}_{m4} = \mathbf{y}_4 = \tan(\beta_2) = \frac{h_{A_2}}{|z_{r,2}|}. \quad (4.37)$$

Théorème 4.3 *Un robot de type (2, 1) décrit par son modèle cinématique (4.4), et son modèle de mesure (4.34)-(4.37) (correspondant à la mesure des angles relatifs entre lui-même et un point d'intérêt et entre lui-même et une cible) peut reconstruire sa posture, l'altitude de la cible et ses commandes en utilisant les estimateurs suivants :*

$$\hat{v} = \mathcal{I} \left(\frac{2h_{A_1} \hat{\mathbf{y}}_{aux2} \hat{\mathbf{y}}_{aux1}}{(1 + \hat{\mathbf{y}}_{aux1}^2) \hat{\mathbf{y}}_{aux2}^2} \right), \quad (4.38)$$

$$\hat{z}_{A_2} = \frac{h_{A_1} \hat{\mathbf{y}}_{aux2} \hat{\mathbf{y}}_{aux1}}{(1 + \hat{\mathbf{y}}_{aux1}^2) \hat{\mathbf{y}}_{aux2}^2} \times \frac{(1 + \hat{\mathbf{y}}_{aux3}^2) \hat{\mathbf{y}}_{aux4}^2}{\hat{\mathbf{y}}_{aux4} \hat{\mathbf{y}}_{aux3}}, \quad (4.39)$$

$$\hat{\Gamma}(t) = \Gamma(t_0) \frac{h_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t_0)}{\hat{\mathbf{y}}_{aux2}(t_0)} - h_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t_0)}{\hat{\mathbf{y}}_{aux4}(t_0)}}{h_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)} - h_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t)}{\hat{\mathbf{y}}_{aux4}(t)}}, \quad (4.40)$$

$$\hat{z}(t) = z_{A_1} - h_{A_1} \Gamma(t) \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)}, \quad (4.41)$$

$$\hat{\omega}_3 = -i \frac{num}{h_{A_1} \hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} \hat{\mathbf{y}}_{aux4}^2 - h_{A_2} \hat{\mathbf{y}}_{aux2}^2 \hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4}}, \quad (4.42)$$

$$\begin{aligned} num &= h_{A_1} \hat{\mathbf{y}}_{aux4}^2 \left(\hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} - \hat{\mathbf{y}}_{aux1} \hat{\mathbf{y}}_{aux2} \right) + \\ & h_{A_2} \hat{\mathbf{y}}_{aux2}^2 \left(\hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4} - \hat{\mathbf{y}}_{aux3} \hat{\mathbf{y}}_{aux4} \right) \\ \hat{\mathbf{y}}_{aux\ell} &= y_{f\ell}, \ell = 1, \dots, 4. \end{aligned} \quad (4.43)$$

Démonstration. La démonstration est la même que celle du théorème 3.5 en remplaçant dans les équations Θ par Γ et v par v_1 . \square

Remarque 4.1 *En observant le théorème 4.3, le lecteur peut remarquer qu'il est impossible en mesurant les angles relatifs entre le robot et un certain nombre de points d'intérêt de faire la distinction entre l'angle d'orientation de la roue β et l'orientation du robot θ . En supposant maintenant que ce robot peut mesurer β , alors il devient possible d'estimer la vitesse angulaire du robot $w_1 = w - w_2$ et $\Theta = \Gamma - B$*

4.2.1.5 Localisation d'un robot de type (1, 1)

Théorème 4.4 *Un robot de type (1, 1) décrit par son modèle cinématique (4.9) et son modèle de mesure (4.12)-(4.15) (correspondant à la mesure des angles relatifs entre lui-même et un point d'intérêt et entre lui-même et une cible) peut reconstruire sa posture, l'altitude de la cible et ses commandes en utilisant les estimateurs suivants :*

$$\hat{v}_2 = \frac{2h_{A_1}\hat{\mathbf{y}}_{aux2}\hat{\mathbf{y}}_{aux1}}{(1 + \hat{\mathbf{y}}_{aux1}^2)\hat{\mathbf{y}}_{aux2}^2}, \quad (4.44)$$

$$\hat{z}_{A_2} = \frac{h_{A_1}\hat{\mathbf{y}}_{aux2}\hat{\mathbf{y}}_{aux1}}{(1 + \hat{\mathbf{y}}_{aux1}^2)\hat{\mathbf{y}}_{aux2}^2} \times \frac{(1 + \hat{\mathbf{y}}_{aux3}^2)\hat{\mathbf{y}}_{aux4}^2}{\hat{\mathbf{y}}_{aux4}\hat{\mathbf{y}}_{aux3}}, \quad (4.45)$$

$$\hat{\Theta}(t) = \Theta(t_0) \frac{h_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t_0)}{\hat{\mathbf{y}}_{aux2}(t_0)} - h_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t_0)}{\hat{\mathbf{y}}_{aux4}(t_0)}}{h_{A_1} \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)} - h_{A_2} \frac{\hat{\mathbf{y}}_{aux3}(t)}{\hat{\mathbf{y}}_{aux4}(t)}}, \quad (4.46)$$

$$\hat{z}(t) = z_{A_1} - h_{A_1} \Theta(t) \frac{\hat{\mathbf{y}}_{aux1}(t)}{\hat{\mathbf{y}}_{aux2}(t)}, \quad (4.47)$$

$$\hat{v}_3 = -i \frac{num}{h_{A_1}\hat{\mathbf{y}}_{aux1}\hat{\mathbf{y}}_{aux2}\hat{\mathbf{y}}_{aux4}^2 - h_{A_2}\hat{\mathbf{y}}_{aux2}^2\hat{\mathbf{y}}_{aux3}\hat{\mathbf{y}}_{aux4}} \quad (4.48)$$

$$num = h_{A_1}\hat{\mathbf{y}}_{aux4}^2 \left(\hat{\mathbf{y}}_{aux1}\hat{\mathbf{y}}_{aux2} - \hat{\mathbf{y}}_{aux1}\hat{\mathbf{y}}_{aux2} \right) + h_{A_2}\hat{\mathbf{y}}_{aux2}^2 \left(\hat{\mathbf{y}}_{aux3}\hat{\mathbf{y}}_{aux4} - \hat{\mathbf{y}}_{aux3}\hat{\mathbf{y}}_{aux4} \right)$$

$$\hat{v} = \frac{|\hat{v}_2 + L\hat{v}_3|}{L} \quad (4.49)$$

$$\hat{B} = \frac{\hat{v}_2 + L\hat{v}_3}{|\hat{v}_2 + L\hat{v}_3|} \quad (4.50)$$

$$\hat{\omega} = -i \times \frac{\hat{B}}{\hat{B}} \quad (4.51)$$

$$\hat{\mathbf{y}}_{aux\ell} = y_{f\ell}, \ell = 1, \dots, 4. \quad (4.52)$$

Démonstration. L'estimation des vitesses v_2 et v_3 , de la posture et de l'altitude de la cible se démontre de la même façon que le théorème 3.5.

Calculons ensuite $v_2 + Lv_3$

$$v_2 + Lv_3 = iLI(B)v + LR(B)v \quad (4.53)$$

$$= LBv \quad (4.54)$$

$$|v_2 + Lv_3| = Lv \quad (4.55)$$

Cette équation permet, en remplaçant v_2 et v_3 par leur expression estimée, d'obtenir l'estimateur de la vitesse linéaire v du robot.

En utilisant le même raisonnement il est possible d'obtenir l'expression de B et donc finalement celle de ω \square

4.2.2 Un point d'intérêt avec boussole (mesures : azimut, élévation et θ)

4.2.2.1 Rappel du modèle de mesure

Nous supposons un robot mesurant les angles relatifs α et β entre lui et un point d'intérêt et son orientation θ à l'aide par exemple d'une boussole. Le modèle de mesure est le suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Theta} \quad (4.56)$$

$$\mathbf{y}_{m2} = \frac{h_{A_1}}{|z_{r,1}|} \quad (4.57)$$

$$\mathbf{y}_{m3} = \Theta \quad (4.58)$$

À partir de ce modèle de mesure nous construisons les deux sorties auxiliaires suivantes (à condition que $h_{A_1} \neq 0$) :

$$\hat{\mathbf{y}}_{aux1} = z_{r,1} \quad (4.59)$$

$$\hat{\mathbf{y}}_{aux2} = \Theta \quad (4.60)$$

4.2.2.2 Localisation d'un robot de type $(2, 0)$

Nous avons montré qu'un tel robot est localisable et un estimateur d'état et de commande a été donné dans la section 3.3.4.2. Rappelons ici le résultat que nous avons obtenu :

Théorème 4.5 (voir le théorème 3.7) *Un robot de type $(2, 0)$ qui mesure les angles relatifs entre lui et un point d'intérêt et qui mesure en plus son orientation (Equations (4.56)-(4.58) peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\hat{z} = z_{A_1} - \hat{\mathbf{y}}_{aux1}, \quad (4.61)$$

$$\hat{\Theta} = \hat{\mathbf{y}}_{aux2} \quad (4.62)$$

$$\hat{v} = -\frac{\hat{\mathbf{y}}_{aux1}}{\hat{\mathbf{y}}_{aux2}}, \quad (4.63)$$

$$\hat{\omega} = \frac{-i\hat{\mathbf{y}}_{aux2}}{\hat{\mathbf{y}}_{aux2}}. \quad (4.64)$$

$$\{\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}\} = \left\{ \frac{\mathbf{y}_{f1}\mathbf{y}_{f3}h_{A_1}}{\mathbf{y}_{f2}}, \mathbf{y}_{f3} \right\} \quad (4.65)$$

4.2.2.3 Localisation d'un robot de type $(3, 0)$

Théorème 4.6 *Un robot de type $(3, 0)$ décrit par son modèle cinématique (4.2) et son modèle de mesure (4.56)-(4.58) (correspondant à la mesure des angles relatifs entre lui-même et un point d'intérêt et à la mesure de son orientation) peut reconstruire sa posture*

et ses commandes en utilisant les estimateurs suivants :

$$\hat{z} = z_{A_1} - \hat{\mathbf{y}}_{aux1}, \quad (4.66)$$

$$\hat{\Theta} = \hat{\mathbf{y}}_{aux2} \quad (4.67)$$

$$\hat{v}_1 = \mathcal{R} \left(-\frac{\hat{\mathbf{y}}_{aux1}}{\hat{\mathbf{y}}_{aux2}} \right), \quad (4.68)$$

$$\hat{v}_2 = \mathcal{I} \left(-\frac{\hat{\mathbf{y}}_{aux1}}{\hat{\mathbf{y}}_{aux2}} \right), \quad (4.69)$$

$$\hat{\omega} = \frac{-i\hat{\mathbf{y}}_{aux2}}{\hat{\mathbf{y}}_{aux2}}. \quad (4.70)$$

$$\{\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}\} = \left\{ \frac{\mathbf{y}_{f1}\mathbf{y}_{f3}h_{A_1}}{\mathbf{y}_{f2}}, \mathbf{y}_{f3} \right\} \quad (4.71)$$

Démonstration. La démonstration est la même que celle du théorème 3.7 en utilisant le changement de variable $v = v_1 + iv_2$ \square

4.2.2.4 Localisation d'un robot de type (2, 1)

Le modèle de mesure est décrit par les équations (4.72)-(4.74) (nous supposons que le robot est capable de mesurer l'orientation de sa roue orientable à l'aide par exemple d'une codeuse montée sur l'axe de rotation verticale de la roue).

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Gamma} \quad (4.72)$$

$$\mathbf{y}_{m2} = \frac{h_{A_1}}{|z_{r,1}|} \quad (4.73)$$

$$\mathbf{y}_{m3} = \Gamma \quad (4.74)$$

et les sorties auxiliaires suivantes :

$$\hat{\mathbf{y}}_{aux1} = z_{r,1} \quad (4.75)$$

$$\hat{\mathbf{y}}_{aux2} = \Gamma \quad (4.76)$$

Théorème 4.7 *Un robot de type (2, 1) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.72)-(4.74) (correspondant à la mesure des angles relatifs entre lui-même et un point d'intérêt et à la mesure de son orientation) peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\hat{z} = z_{A_1} - \hat{\mathbf{y}}_{aux1}, \quad (4.77)$$

$$\hat{\Gamma} = \hat{\mathbf{y}}_{aux2} \quad (4.78)$$

$$\hat{v}_1 = \mathcal{I} \left(-\frac{\hat{\mathbf{y}}_{aux1}}{\hat{\mathbf{y}}_{aux2}} \right), \quad (4.79)$$

$$\hat{\omega} = \frac{-i\hat{\mathbf{y}}_{aux2}}{\hat{\mathbf{y}}_{aux2}}. \quad (4.80)$$

$$\{\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}\} = \left\{ \frac{\mathbf{y}_{f1}\mathbf{y}_{f3}h_{A_1}}{\mathbf{y}_{f2}}, \mathbf{y}_{f3} \right\} \quad (4.81)$$

Démonstration. La démonstration est la même que celle du théorème 3.7 en estimant Γ au lieu d'estimer Θ . \square

Remarque 4.2 *Le lecteur peut remarquer qu'il est impossible en mesurant les angles relatifs entre le robot et un certain nombre de points d'intérêt de faire la distinction entre l'orientation de la roue β et l'orientation du robot θ . En supposant maintenant que ce robot peut mesurer β alors il devient possible d'estimer la vitesse angulaire du robot $w_1 = w_3 - w_2$ et son orientation θ à partir de $\Theta = \Gamma - B$*

4.2.2.5 Localisation d'un robot de type (1, 1)

Théorème 4.8 *Un robot de type (1, 1) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.56)-(4.58) peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\hat{z} = z_{A_1} - \hat{\mathbf{y}}_{aux1}, \quad (4.82)$$

$$\hat{\Theta} = \hat{\mathbf{y}}_{aux2} \quad (4.83)$$

$$\hat{v}_2 = -\frac{\hat{\mathbf{y}}_{aux1}}{\hat{\mathbf{y}}_{aux2}}, \quad (4.84)$$

$$\hat{v}_3 = \frac{-i\hat{\mathbf{y}}_{aux2}}{\hat{\mathbf{y}}_{aux2}}. \quad (4.85)$$

$$\hat{v} = \frac{|\hat{v}_2 + L\hat{v}_3|}{L} \quad (4.86)$$

$$\hat{B} = \frac{\hat{v}_2 + L\hat{v}_3}{|\hat{v}_2 + L\hat{v}_3|} \quad (4.87)$$

$$\hat{\omega} = -i \times \frac{\hat{B}}{B} \quad (4.88)$$

$$\{\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}\} = \left\{ \frac{\mathbf{y}_{f1}\mathbf{y}_{f3}h_{A_1}}{\mathbf{y}_{f2}}, \mathbf{y}_{f3} \right\} \quad (4.89)$$

Démonstration. La démonstration est la même que celle du théorème 3.7. L'estimation de v , B et ω est issue du changement de variable effectué. \square

4.2.3 Deux points d'intérêt (mesures : deux élévations)

4.2.3.1 Rappel du modèle de mesure

Nous supposons un robot mesurant les élévations entre lui et deux points d'intérêt. Le modèle de mesure est le suivant :

$$\mathbf{y}_{m1} = \frac{h_{A_1}}{|z_{r,1}|} \quad (4.90)$$

$$\mathbf{y}_{m2} = \frac{h_{A_2}}{|z_{r,2}|} \quad (4.91)$$

$$(4.92)$$

À partir de ce modèle de mesure il est possible de construire les deux sorties auxiliaires suivantes (à condition que $h_{A_\ell} \neq 0$, $\ell = 1, 2$) :

$$\hat{\mathbf{y}}_{aux1} = \frac{1}{2}|z_{r,1}|^2 \quad (4.93)$$

$$\hat{\mathbf{y}}_{aux2} = \frac{1}{2}|z_{r,2}|^2 \quad (4.94)$$

4.2.3.2 Localisation d'un robot de type (2, 0)

Nous avons montré qu'un tel robot est localisable et un estimateur d'état et de commande a été donné dans la section 3.3.5.2. Rappelons ici le résultat que nous avons obtenu :

Théorème 4.9 *Un robot unicycle mesurant les élévations relatives entre lui-même et deux points d'intérêt (Équations (4.90)-(4.91) peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\hat{z} = z_{A_1} + z_{A_{12}} \left(f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm \sqrt{r_1^2 - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})} \right), \quad (4.95)$$

$$\hat{\Theta} = \frac{\hat{z}}{|\hat{z}|}. \quad (4.96)$$

$$\hat{v} = |\hat{z}|, \quad (4.97)$$

$$\hat{\omega} = -\frac{i}{2} \left(\frac{\hat{z}}{|\hat{z}|} - \frac{\hat{z}\hat{z}}{|\hat{z}|^2} \right). \quad (4.98)$$

$$\hat{z} = z_{A_{12}} \left(\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm i \frac{\hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})}{\sqrt{2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})}} \right),$$

$$\hat{z} = z_{A_{12}} \left(\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm \left(+ic_{A_{12}} \left(\frac{num}{denum} \right) - ic_{A_{12}} \left(\frac{(\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2})^2}{denum} \right) \right) \right),$$

$$num = (2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})) \left(\hat{\mathbf{y}}_{aux1} - \frac{\hat{\mathbf{y}}_{aux1}^2}{2\hat{\mathbf{y}}_{aux1}} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \right)$$

$$denum = (2\hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})^2)^{\frac{3}{2}},$$

$$f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}} + \frac{d_{A_{12}}}{2}, \quad (4.99)$$

$$\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\dot{\hat{\mathbf{y}}}_{aux1} - \dot{\hat{\mathbf{y}}}_{aux2}}{d_{A_{12}}}, \quad (4.100)$$

$$\ddot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\ddot{\hat{\mathbf{y}}}_{aux1} - \ddot{\hat{\mathbf{y}}}_{aux2}}{d_{A_{12}}}, \quad (4.101)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|, \quad (4.102)$$

$$\mathbf{y}_{auxl} = \frac{1}{2} \left(\frac{h_{A_l}}{\mathbf{y}_{fl}} \right)^2, \quad l = 1, 2. \quad (4.103)$$

4.2.3.3 Localisation d'un robot de type (3, 0)

Théorème 4.10 *Un robot de type (3, 0) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.90)-(4.91) (correspondant à la mesure de l'élévation relative entre lui-même et deux points d'intérêt) peut reconstruire sa posture en utilisant l'estimateur*

suivant :

$$\hat{z} = z_{A_1} + z_{A_{12}} \left(f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm \sqrt{2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})} \right) \quad (4.104)$$

$$\hat{\Theta} = \frac{\hat{z}}{v_1 + iv_2}, \quad (4.105)$$

$$f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}} + \frac{d_{A_{12}}}{2}, \quad (4.106)$$

$$\hat{\dot{z}} = z_{A_{12}} \dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \quad (4.107)$$

$$\pm iz_{A_{12}} \left(\frac{\hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})}{\sqrt{2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})}} \right), \quad (4.108)$$

$$\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) = \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}}. \quad (4.109)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.110)$$

$$\hat{\mathbf{y}}_{auxl} = \frac{1}{2} \left(\frac{h_{A_l}}{\mathbf{y}_{fl}} \right)^2, \quad l = 1, 2. \quad (4.111)$$

Démonstration. Contrairement au robot de type (2, 0) les commandes ($v = v_1 + iv_2$ et ω) doivent être connues pour pouvoir estimer l'orientation du robot. En effet, la commande étant un complexe on ne peut différencier Θ et les commandes dans le modèle. La démonstration est ensuite équivalente à celle donnée pour un robot de type (2, 0) avec $v = v_1 + iv_2$ (voir la démonstration du théorème 3.9) \square

4.2.3.4 Localisation d'un robot de type (2, 1)

Théorème 4.11 *Un robot de type (2, 1) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.90)-(4.91) (correspondant à la mesure de l'élévation relative entre lui-même et deux points d'intérêt) peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\hat{z} = z_{A_1} + z_{A_{12}} \left(f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm \sqrt{2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})} \right), \quad (4.112)$$

$$\hat{\Gamma} = -i \times \frac{\hat{\dot{z}}}{|\hat{\dot{z}}|}. \quad (4.113)$$

$$\hat{v} = |\hat{\dot{z}}|, \quad (4.114)$$

$$\hat{\omega} = \frac{1}{2} \left(\frac{\hat{\dot{z}}}{\hat{z}} - \frac{\hat{\dot{z}}\hat{\dot{z}}}{|\hat{\dot{z}}|^2} \right). \quad (4.115)$$

$$\begin{aligned}
\hat{z} &= z_{A_{12}} \left(\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm i \frac{num_1}{\sqrt{2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})}} \right), \\
num_1 &= \hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \\
\hat{z} &= z_{A_{12}} \left(\ddot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \pm \right. \\
&\quad \left(+ic_{A_{12}} \left(\frac{num_2}{(2\hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})^2)^{\frac{3}{2}}} \right) \right. \\
&\quad \left. \left. - ic_{A_{12}} \left(\frac{(\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2})^2}{(2\hat{\mathbf{y}}_{aux1} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})^2)^{\frac{3}{2}}} \right) \right) \right), \\
num_2 &= (2\hat{\mathbf{y}}_{aux1} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2})) \\
&\quad \times \left(\hat{\mathbf{y}}_{aux1} - \frac{\hat{\mathbf{y}}_{aux1}^2}{2\hat{\mathbf{y}}_{aux1}} - f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \ddot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \right) \\
f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) &= \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}} + \frac{d_{A_{12}}}{2}, \quad (4.116) \\
\dot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) &= \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}}, \quad (4.117) \\
\ddot{f}(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) &= \frac{\hat{\mathbf{y}}_{aux1} - \hat{\mathbf{y}}_{aux2}}{d_{A_{12}}}. \quad (4.118)
\end{aligned}$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.119)$$

$$\hat{\mathbf{y}}_{auxl} = \frac{1}{2} \left(\frac{h_{A_l}}{\mathbf{y}_{fl}} \right)^2, \quad l = 1, 2. \quad (4.120)$$

Démonstration. La démonstration est la même que celle du théorème 3.9 en estimant Γ au lieu d'estimer Θ . \square

Remarque 4.3 *Le lecteur peut remarquer qu'il est impossible en mesurant les angles relatifs entre le robot et un certain nombre de points d'intérêt de faire la distinction entre l'orientation de la roue β et l'orientation du robot θ . En supposant maintenant que ce robot peut mesurer β alors il devient possible d'estimer la vitesse de rotation du robot $w_1 = w - w_2$ et son orientation θ à partir de $\Theta = \Gamma - B$.*

4.2.3.5 Localisation d'un robot de type (1, 1)

Théorème 4.12 *Un robot de type (2, 1) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.90)-(4.91) (correspondant à la mesure de l'élévation relative entre lui-même et deux points d'intérêt) peut reconstruire sa posture et ses commandes en utilisant les estimateurs suivants :*

$$\widehat{z} = z_{A_1} + z_{A_{12}} \left(f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) \pm \sqrt{2\widehat{\mathbf{y}}_{aux1} - f^2(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2})} \right), \quad (4.121)$$

$$\widehat{\Theta} = -i \times \frac{\widehat{\dot{z}}}{|\widehat{\dot{z}}|}. \quad (4.122)$$

$$\widehat{v}_2 = i|\widehat{\dot{z}}|, \quad (4.123)$$

$$\widehat{v}_3 = \frac{1}{2} \left(\frac{\widehat{\dot{z}}}{\widehat{\dot{z}}} - \frac{\widehat{\dot{z}}\overline{\widehat{\dot{z}}}}{|\widehat{\dot{z}}|^2} \right). \quad (4.124)$$

$$\widehat{v} = \frac{|\widehat{v}_2 + L\widehat{v}_3|}{L} \quad (4.125)$$

$$\widehat{B} = \frac{\widehat{v}_2 + L\widehat{v}_3}{|\widehat{v}_2 + L\widehat{v}_3|} \quad (4.126)$$

$$\widehat{\dot{z}} = z_{A_{12}} \left(\dot{f}(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) \pm i \frac{num_1}{\sqrt{2\widehat{\mathbf{y}}_{aux1} - f^2(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2})}} \right),$$

$$num_2 = \widehat{\mathbf{y}}_{aux1} - f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) \dot{f}(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2})$$

$$\begin{aligned} \widehat{\ddot{z}} = & z_{A_{12}} \left(\ddot{f}(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) \pm \right. \\ & \left(+ic_{A_{12}} \left(\frac{num_2}{(2\widehat{\mathbf{y}}_{aux1} - f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2})^2)^{\frac{3}{2}}} \right) \right. \\ & \left. \left. - ic_{A_{12}} \left(\frac{(\widehat{\mathbf{y}}_{aux1} - \widehat{\mathbf{y}}_{aux2})^2}{(2\widehat{\mathbf{y}}_{aux1} - f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2})^2)^{\frac{3}{2}}} \right) \right) \right), \end{aligned}$$

$$\begin{aligned} num_2 = & (2\widehat{\mathbf{y}}_{aux1} - f^2(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2})) \\ & \times \left(\widehat{\mathbf{y}}_{aux1} - \frac{\widehat{\mathbf{y}}_{aux1}^2}{2\widehat{\mathbf{y}}_{aux1}} - f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) \ddot{f}(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) \right) \end{aligned}$$

$$f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) = \frac{\widehat{\mathbf{y}}_{aux1} - \widehat{\mathbf{y}}_{aux2}}{d_{A_{12}}} + \frac{d_{A_{12}}}{2}, \quad (4.127)$$

$$\dot{f}(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) = \frac{\widehat{\dot{\mathbf{y}}}_{aux1} - \widehat{\dot{\mathbf{y}}}_{aux2}}{d_{A_{12}}}, \quad (4.128)$$

$$\ddot{f}(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) = \frac{\widehat{\ddot{\mathbf{y}}}_{aux1} - \widehat{\ddot{\mathbf{y}}}_{aux2}}{d_{A_{12}}}. \quad (4.129)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.130)$$

$$\widehat{\mathbf{y}}_{aux\ell} = \frac{1}{2} \left(\frac{h_{A_\ell}}{\mathbf{y}_{f\ell}} \right)^2, \quad \ell = 1, 2. \quad (4.131)$$

Démonstration. La démonstration est la même que pour le théorème 3.9. L'estimation de v , B et ω est issue du changement de variable effectué. \square

4.2.4 Deux points d'intérêt (mesures : deux azimuts)

4.2.4.1 Rappel du modèle de mesure

Nous sommes dans le cas d'un robot qui observe deux points d'intérêt et qui mesure uniquement les azimuts entre lui-même et ces deux points d'intérêt. Le modèle de mesure est donc le suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}| \Theta} \quad (4.132)$$

$$\mathbf{y}_{m2} = \frac{z_{r,2}}{|z_{r,2}| \Theta} \quad (4.133)$$

Nous prenons les deux sorties auxiliaires suivantes :

$$\Theta \mathbf{y}_\ell = \frac{z_{r,\ell}}{|z_{r,\ell}|}, \quad \ell = 1, 2 \quad (4.134)$$

4.2.4.2 Localisation d'un robot de type (2, 0)

Nous avons montré qu'un tel robot est localisable et un estimateur d'état et de commande a été donné dans la section 3.3.5.2. Rappelons ici le résultat que nous avons obtenu :

Théorème 4.13 (voir le théorème 3.11) *Un robot de type (2, 0) qui mesure les azimuts relatifs entre lui et deux points d'intérêt (Équations (4.132)-(4.133)) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned} \hat{z} &= z_{A_1} + z_{A_{12}} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \\ &\pm z_{A_{12}} \left(i \sqrt{\frac{(1 - \hat{\mathbf{y}}_{aux1}^2)^2}{(i\omega \hat{\mathbf{y}}_1 + \hat{\mathbf{y}}_1)^2}} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \right) \end{aligned} \quad (4.135)$$

$$\hat{\Theta} = \frac{z_{A_1} - z}{|z_{A_1} - z| \times \hat{\mathbf{y}}_{aux1}}, \quad (4.136)$$

$$\begin{aligned} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) &= \frac{v^2}{8d_{A_{12}}} \left(\frac{(1 - \hat{\mathbf{y}}_{aux1}^2)^2}{(i\omega \hat{\mathbf{y}}_1 + \hat{\mathbf{y}}_1)^2} - \frac{(1 - \hat{\mathbf{y}}_{aux2}^2)^2}{(i\omega \hat{\mathbf{y}}_2 + \hat{\mathbf{y}}_2)^2} \right) \\ &+ \frac{d_{A_{12}}}{2}. \end{aligned} \quad (4.137)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.138)$$

4.2.4.3 Localisation d'un robot de type (3, 0)

Théorème 4.14 *Un robot de type (3, 0) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.132)-(4.133) (correspondant à la mesure des azimuts relatifs entre lui-même et deux points d'intérêt) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned} \hat{z} &= z_{A_1} + z_{A_{12}} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \\ &\pm z_{A_{12}} \left(i \sqrt{|z_{r,1}|^2} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \right), \end{aligned} \quad (4.139)$$

$$\hat{\Theta} = \frac{z_{A_1} - z}{|z_{A_1} - z| \times \hat{\mathbf{y}}_{aux1}}, \quad (4.140)$$

$$|z_{r,\ell}| = \frac{(v_1 - iv_2)\mathbf{y}_{aux\ell}^2 - (v_1 + iv_2)}{2(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell})}. \quad (4.141)$$

$$f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\dot{\mathbf{y}}}_{aux1}, \widehat{\dot{\mathbf{y}}}_{aux2}) = \frac{|z_{r,1}|^2 - |z_{r,2}|^2 + d_{A12}^2}{2d_{A12}} \quad (4.142)$$

$$z_{A12} = \frac{z_{A2} - z_{A1}}{d_{A12}}, \quad d_{A12} = |z_{A2} - z_{A1}|. \quad (4.143)$$

Démonstration. Calculons les dérivées temporelles des sorties auxiliaires :

$$\dot{\Theta}\mathbf{y}_{aux\ell} + \Theta\dot{\mathbf{y}}_{aux\ell} = \frac{\dot{z}_{r,\ell}|z_{r,\ell}| - z_{r,\ell}\frac{d|z_{r,\ell}|}{dt}}{|z_{r,\ell}|^2}. \quad (4.144)$$

Or

$$\frac{d|z_{r,\ell}|}{dt} = \frac{z_{r,\ell}\bar{\dot{z}}_{r,\ell} + \bar{z}_{r,\ell}\dot{z}_{r,\ell}}{2|z_{r,\ell}|}, \quad (4.145)$$

avec $\dot{z}_{r,\ell} = -\dot{z} = -(v_1 + iv_2)\Theta$ et $\bar{\dot{z}}_{r,\ell} = -(v_1 - iv_2)\bar{\Theta} = -\bar{v}\Theta^{-1}$, la relation algébrique (4.144) donne (en utilisant (4.2)) :

$$(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell}) = \frac{-(v_1 + iv_2)|z_{r,\ell}| - z_{r,\ell} \times frac}{|z_{r,\ell}|^2}, \quad (4.146)$$

$$frac = \left[\frac{-z_{r,\ell}(v_1 - iv_2)\Theta^{-2} - \bar{z}_{r,\ell}(v_1 + iv_2)}{2|z_{r,\ell}|} \right]$$

$$(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell}) = -\frac{v_1 + iv_2}{2|z_{r,\ell}|} + \frac{(v_1 - iv_2)\mathbf{y}_{aux\ell}^2}{2|z_{r,\ell}|} \quad (4.147)$$

Nous obtenons finalement

$$|z_{r,\ell}| = \frac{(v_1 - iv_2)\mathbf{y}_{aux\ell}^2 - (v_1 + iv_2)}{2(i\omega\mathbf{y}_{aux\ell} + \dot{\mathbf{y}}_{aux\ell})}. \quad (4.148)$$

Le résultat est alors donné directement comme étant l'intersection de deux cercles de rayon $|z_{r,\ell}|$ (voir l'annexe A.4). \square

4.2.4.4 Localisation d'un robot de type (2, 1)

En utilisant le modèle de mesure suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Gamma} \quad (4.149)$$

$$\mathbf{y}_{m2} = \frac{z_{r,2}}{|z_{r,2}|\Gamma} \quad (4.150)$$

Nous prenons les deux sorties auxiliaires suivantes :

$$\Gamma\mathbf{y}_\ell = \frac{z_{r,\ell}}{|z_{r,\ell}|}, \quad \ell = 1, 2 \quad (4.151)$$

Théorème 4.15 *Un robot de type (2, 1) décrit par son modèle cinématique (4.4) et son modèle de mesure (4.149)-(4.150) (correspondant à la mesure des azimuts relatifs entre lui-même et deux points d'intérêt) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned}\hat{z} &= z_{A_1} + z_{A_{12}} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \\ &\pm z_{A_{12}} \left(i \sqrt{\frac{(1 - \hat{\mathbf{y}}_{aux1}^2)^2}{(i\omega\hat{\mathbf{y}}_1 + \hat{\mathbf{y}}_1)^2}} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \right) \quad (4.152) \\ \hat{\Theta} &= \frac{z_{A_1} - z}{|z_{A_1} - z| \times \hat{\mathbf{y}}_{aux1}}, \quad (4.153)\end{aligned}$$

$$\begin{aligned}f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) &= \frac{-v^2}{8d_{A_{12}}} (frac_1 - frac_2) \\ &+ \frac{d_{A_{12}}}{2}, \quad (4.154)\end{aligned}$$

$$\begin{aligned}frac_1 &= \frac{(1 - \hat{\mathbf{y}}_{aux1}^2)^2}{(i(\omega_2 + \omega_3)\hat{\mathbf{y}}_1 + \hat{\mathbf{y}}_1)^2}, \\ frac_2 &= \frac{(1 - \hat{\mathbf{y}}_{aux2}^2)^2}{(i(\omega_2 + \omega_3)\hat{\mathbf{y}}_2 + \hat{\mathbf{y}}_2)^2}, \quad (4.155)\end{aligned}$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.156)$$

Démonstration. La démonstration est la même que pour le théorème 3.11 en estimant Γ au lieu d'estimer Θ . \square

Remarque 4.4 *Le lecteur peut remarquer qu'il est impossible en mesurant les angles relatifs entre le robot et un certain nombre de points d'intérêt de faire la distinction entre l'orientation de la roue β et l'orientation du robot θ . En supposant maintenant que ce robot peut mesurer β alors il devient possible d'estimer son orientation θ à partir de $\Theta = \Gamma - B$.*

4.2.4.5 Localisation d'un robot de type (1, 1)

En supposant B connus :

Théorème 4.16 *Un robot de type (1, 1) décrit par son modèle cinématique (4.9) et son modèle de mesure (4.132)-(4.133) (correspondant à la mesure des azimuts relatifs entre lui-même et deux points d'intérêt) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned}\hat{z} &= z_{A_1} + z_{A_{12}} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \\ &\pm z_{A_{12}} \left(i \sqrt{\frac{(1 - \hat{\mathbf{y}}_{aux1}^2)^2}{(i\omega\hat{\mathbf{y}}_1 + \hat{\mathbf{y}}_1)^2}} - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}) \right) \quad (4.157) \\ \hat{\Theta} &= \frac{z_{A_1} - z}{|z_{A_1} - z| \times \hat{\mathbf{y}}_{aux1}}, \quad (4.158)\end{aligned}$$

$$f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) = \frac{(iL\mathcal{I}(B)v)^2}{8d_{A_{12}}} (frac_1 - frac_2) + \frac{d_{A_{12}}}{2}, \quad (4.159)$$

$$frac_1 = \frac{(1 - \widehat{\mathbf{y}}_{aux1}^2)^2}{(i\mathcal{R}(B)v\widehat{\mathbf{y}}_1 + \widehat{\mathbf{y}}_1)^2}$$

$$frac_2 = \frac{(1 - \widehat{\mathbf{y}}_{aux2}^2)^2}{(\mathcal{R}(B)v\widehat{\mathbf{y}}_2 + \widehat{\mathbf{y}}_2)^2} \quad (4.160)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.161)$$

Démonstration. La démonstration est la même que pour le théorème 3.11. \square

4.2.5 Deux points d'intérêt (mesures : une élévation et un azimut)

4.2.5.1 Rappel du modèle de mesure

Nous sommes dans le cas d'un robot qui observe deux points d'intérêt et qui mesure l'azimut par rapport à l'un et l'élévation par rapport à l'autre. Le modèle de mesure est donc le suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Theta} \quad (4.162)$$

$$\mathbf{y}_{m2} = \frac{h_{A_2}}{|z_{r,2}|} \quad (4.163)$$

À partir de HM5, quand $h_{A_\ell} \neq 0, \ell = 1, 2$ il est possible de prendre les sorties auxiliaires suivantes :

$$\mathbf{y}_{aux1} = |z_{r,1}|, \quad (4.164)$$

$$\mathbf{y}_{aux2} = \frac{z_{r,2}}{\Theta |z_{r,2}|}. \quad (4.165)$$

4.2.5.2 Localisation d'un robot de type (2, 0)

Nous avons montré qu'un tel robot est localisable et un estimateur d'état et de commande a été donné dans la section 3.3.7. Rappelons ici le résultat que nous avons obtenu :

Théorème 4.17 (voir le théorème 3.13) *Un robot de type (2, 0) qui mesure l'élévation relative entre lui-même et un point d'intérêt et l'azimut relatif entre lui-même et un autre point d'intérêt (Equations (4.162)-(4.163)) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\widehat{z} = z_{A_1} + z_{A_{12}} \left(f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\mathbf{y}}_{aux2}) \right) \pm z_{A_{12}} i \sqrt{\widehat{\mathbf{y}}_{aux1}^2 - f^2(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\mathbf{y}}_{aux2})}, \quad (4.166)$$

$$\widehat{\Theta} = \frac{z_{A_2} - z}{\widehat{\mathbf{y}}_2(|z_{A_{aux2}} - z|)} \quad (4.167)$$

$$f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\mathbf{y}}_{aux2}) = \frac{\widehat{\mathbf{y}}_{aux1}^2}{2d_{A12}} - \frac{v^2(1 - \widehat{\mathbf{y}}_{aux2}^2)^2}{8d_{A12}(i\omega\widehat{\mathbf{y}}_{aux2} + \widehat{\mathbf{y}}_{aux2})^2} + \frac{d_{A12}}{2} \quad (4.168)$$

$$z_{A12} = \frac{z_{A2} - z_{A1}}{d_{A12}}, \quad d_{A12} = |z_{A2} - z_{A1}|. \quad (4.169)$$

4.2.5.3 Localisation d'un robot de type (3, 0)

Théorème 4.18 *Un robot de type (3, 0) décrit par son modèle cinématique (4.2) et son modèle de mesure (4.162)-(4.163) (correspondant à la mesure de l'azimut par rapport à un point d'intérêt et l'élévation par rapport à un autre point d'intérêt) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\widehat{z} = z_{A1} + z_{A12}f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) \pm z_{A12} \left(i\sqrt{|z_{r,1}|^2 - f^2(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2})} \right), \quad (4.170)$$

$$\widehat{\Theta} = \frac{z_{A1} - z}{|z_{A1} - z| \times \widehat{\mathbf{y}}_{aux1}}, \quad (4.171)$$

$$|z_{r,1}| = \mathbf{y}_{aux1} \quad (4.172)$$

$$|z_{r,2}| = \frac{(v_1 - iv_2)\mathbf{y}_{aux2}^2 - (v_1 + iv_2)}{2(i\omega\mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2})}. \quad (4.173)$$

$$f(\widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}, \widehat{\mathbf{y}}_{aux1}, \widehat{\mathbf{y}}_{aux2}) = \frac{|z_{r,1}|^2 - |z_{r,2}|^2 + d_{A12}^2}{2d_{A12}} \quad (4.174)$$

$$z_{A12} = \frac{z_{A2} - z_{A1}}{d_{A12}}, \quad d_{A12} = |z_{A2} - z_{A1}|. \quad (4.175)$$

Démonstration. Calculons les dérivées temporelles de la sortie auxiliaire \mathbf{y}_{aux2} :

$$\dot{\Theta}\mathbf{y}_{aux2} + \Theta\dot{\mathbf{y}}_{aux2} = \frac{\dot{z}_{r,2}|z_{r,2}| - z_{r,2}\frac{d|z_{r,2}|}{dt}}{|z_{r,2}|^2}. \quad (4.176)$$

Or

$$\frac{d|z_{r,2}|}{dt} = \frac{z_{r,2}\bar{\dot{z}}_{r,2} + \bar{z}_{r,2}\dot{z}_{r,2}}{2|z_{r,2}|}, \quad (4.177)$$

avec $\dot{z}_{r,2} = -\dot{z} = -(v_1 + iv_2)\Theta$ et $\bar{\dot{z}}_{r,2} = -(v_1 - iv_2)\bar{\Theta} = -v\Theta^{-1}$,

La relation algébrique (4.176) donne (en utilisant (5.4)) :

$$(i\omega\mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2}) = \frac{-(v_1 + iv_2)|z_{r,2}| - z_{r,2} \times frac}{|z_{r,2}|^2}, \quad (4.178)$$

$$frac = \left[\frac{-z_{r,2}(v_1 - iv_2)\Theta^{-2} - \bar{z}_{r,2}(v_1 + iv_2)}{2|z_{r,2}|} \right]$$

$$(i\omega\mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2}) = -\frac{v_1 + iv_2}{2|z_{r,2}|} + \frac{(v_1 - iv_2)\mathbf{y}_{aux2}^2}{2|z_{r,2}|} \quad (4.179)$$

Nous obtenons finalement

$$|z_{r,2}| = \frac{(v_1 - iv_2)\mathbf{y}_{aux2}^2 - (v_1 + iv_2)}{2(i\omega\mathbf{y}_{aux2} + \dot{\mathbf{y}}_{aux2})}. \quad (4.180)$$

De plus nous avons $|z_{r,1}| = \mathbf{y}_{aux1}$.

Le résultat est alors donné directement comme étant l'intersection de deux cercles de rayon $|z_{r,\ell}|$ (voir l'annexe A.4). \square

4.2.5.4 Localisation d'un robot de type (2, 1)

En prenant le modèle de mesure suivant :

$$\mathbf{y}_{m1} = \frac{z_{r,1}}{|z_{r,1}|\Gamma} \quad (4.181)$$

$$\mathbf{y}_{m2} = \frac{h_{A_2}}{|z_{r,2}|} \quad (4.182)$$

pour $h_{A_\ell} \neq 0, \ell = 1, 2$, il est possible de prendre les sorties auxiliaires suivantes :

$$\mathbf{y}_1 = |z_{r,1}|, \quad (4.183)$$

$$\mathbf{y}_2 = \frac{z_{r,2}}{\Gamma |z_{r,2}|}. \quad (4.184)$$

Théorème 4.19 *Un robot de type (2, 1) décrit par son modèle cinématique (4.4) et son modèle de mesure (4.181)-(4.182) (correspondant à la mesure de l'azimut par rapport à un point d'intérêt et l'élévation par rapport à un autre point d'intérêt) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned} \hat{z} &= z_{A_1} + z_{A_{12}} \left(f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux2}) \right) \\ &\pm z_{A_{12}} i \sqrt{\hat{\mathbf{y}}_{aux1}^2 - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux2})}, \end{aligned} \quad (4.185)$$

$$\hat{\Theta} = \frac{z_{A_2} - z}{\hat{\mathbf{y}}_2(|z_{A_{aux2}} - z|)} \quad (4.186)$$

$$\begin{aligned} f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux2}) &= \frac{\hat{\mathbf{y}}_{aux1}^2}{2d_{A_{12}}} - \frac{-v^2(1 - \hat{\mathbf{y}}_{aux2}^2)^2}{8d_{A_{12}}(i(\omega_1 + \omega_2)\hat{\mathbf{y}}_{aux2} + \dot{\hat{\mathbf{y}}}_{aux2})^2} \\ &+ \frac{d_{A_{12}}}{2} \end{aligned} \quad (4.187)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.188)$$

Démonstration. La démonstration est la même que pour le théorème 3.13 en estimant Γ au lieu d'estimer Θ . \square

Remarque 4.5 *Le lecteur peut remarquer qu'il est impossible en mesurant les angles relatifs entre le robot et un certain nombre de points d'intérêt de faire la distinction entre l'orientation de la roue β et l'orientation du robot θ . En supposant maintenant que ce robot peut mesurer β alors il devient possible d'estimer son orientation θ à partir de $\Theta = \Gamma - B$.*

4.2.5.5 Localisation d'un robot de type (1, 1)

En supposant B connu :

Théorème 4.20 *Un robot de type (1, 1) décrit par son modèle cinématique (4.9) et son modèle de mesure (4.162)-(4.163) (correspondant à la mesure de l'azimut par rapport à un point d'intérêt et l'élévation par rapport à un autre point d'intérêt) peut reconstruire sa posture en utilisant les estimateurs suivants :*

$$\begin{aligned}\hat{z} &= z_{A_1} + z_{A_{12}} \left(f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux2}) \right) \\ &\pm z_{A_{12}} i \sqrt{\hat{\mathbf{y}}_{aux1}^2 - f^2(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux2})},\end{aligned}\quad (4.189)$$

$$\hat{\Theta} = \frac{z_{A_2} - z}{\hat{\mathbf{y}}_2(|z_{A_{aux2}} - z|)} \quad (4.190)$$

$$\begin{aligned}f(\hat{\mathbf{y}}_{aux1}, \hat{\mathbf{y}}_{aux2}, \hat{\mathbf{y}}_{aux2}) &= \frac{\hat{\mathbf{y}}_{aux1}^2}{2d_{A_{12}}} - \frac{(iL\mathcal{L}(B))v^2(1 - \hat{\mathbf{y}}_{aux2}^2)^2}{8d_{A_{12}}(i\mathcal{R}(B)v\hat{\mathbf{y}}_{aux2} + \hat{\mathbf{y}}_{aux2})^2} \\ &+ \frac{d_{A_{12}}}{2}\end{aligned}\quad (4.191)$$

$$z_{A_{12}} = \frac{z_{A_2} - z_{A_1}}{d_{A_{12}}}, \quad d_{A_{12}} = |z_{A_2} - z_{A_1}|. \quad (4.192)$$

Démonstration. La démonstration est la même que pour le théorème 3.13. \square

4.3 RÉSULTATS DE SIMULATION

4.3.1 Paramètres de simulation et cas d'étude

L'efficacité des estimateurs développés dans la section précédente sera testée en simulation. Le lecteur pourra se reporter à la table 3.1 pour se remémorer les différents cas d'étude. Pour chacun des cas d'étude, nous appelons l'implémentation de l'estimateur correspondant NON-A-HMi, $i=1,1BIS,2,3,4,5$.

En fonction des cas d'étude, il est possible d'estimer soit la posture et les commandes, soit uniquement les commandes (se référer à la table 4.1 pour trouver les numéros de théorème correspondant à chaque cas). Dans le cas où les commandes ne peuvent être estimées, nous les supposons connues et elles sont donc considérées comme une entrée de nos algorithmes.

Chacune des simulations est réitérée 200 fois (chaque simulation étant appelée un jeu de données) de manière à pouvoir calculer les paramètres statistiques des erreurs obtenues (notamment moyenne et variance de l'erreur) en suivant la loi des grands nombres. La période d'échantillonnage des mesures est de $T_s = 1/30sec$. Le bruit de mesure additif sur les angles relatifs est réparti de façon uniforme sur l'intervalle $[-0.0017 \ 0.0017]$ [rad]. Pour chaque simulation, les points d'intérêt sont placés de façon aléatoire dans un cube de l'espace préalablement défini.

L'algorithme de dérivation (voir la section 2.2.4) nécessite quelques paramètres de réglages, notamment la longueur de la fenêtre d'intégration que nous fixons à 80 échantillons, les paramètres $(\kappa, \mu) = (0, 0)$ et l'ordre de troncature de la série de Taylor que nous fixons à l'ordre de dérivation souhaité ($N = n$).

TABLE 4.1 – Table récapitulative des théorèmes relatifs à la localisation des robots mobiles à roues

	Type (2, 0)	Type (3, 0)	Type (2, 1)	Type (1, 1)
<u>NON-A-HM1BIS</u>	Théorème 4.1	Théorème 4.2	Théorème 4.3	Théorème 4.4
<u>NON-A-HM2</u>	Théorème 4.5	Théorème 4.6	Théorème 4.7	Théorème 4.8
<u>NON-A-HM3</u>	Théorème 4.9	Théorème 4.10	Théorème 4.11	Théorème 4.12
<u>NON-A-HM4</u>	Théorème 4.13	Théorème 4.14	Théorème 4.15	Théorème 4.16
<u>NON-A-HM5</u>	Théorème 4.17	Théorème 4.18	Théorème 4.19	Théorème 4.20

TABLE 4.2 – Résultat de simulation pour un robot de type (3, 0)

Algorithmes	Résultats				
	erreur moyenne en distance [m]	variance moyenne de l'erreur en distance [m ²]	erreur moyenne sur v [$m.s^{-1}$]	erreur moyenne sur ω [$rad.s^{-1}$]	Temps d'exécution moyen [s]
<u>NON-A-HM1BIS</u>	0.2150	0.0032	X	0.00097	0.1862
<u>NON-A-HM2</u>	0.1284	0.0027	X	0.001710^{-3}	0.0819
<u>NON-A-HM3</u>	0.2714	0.0035	X	X	0.6613
<u>NON-A-HM4</u>	0.5520	0.1586	X	X	0.6634
<u>NON-A-HM5</u>	0.4658	0.0988	X	X	0.6651

4.3.2 Résultats numériques

Les tables 4.2, 4.3 et 4.4 donnent les résultats pour les robots de type (3, 0), (2, 1) et (1, 1). Dans le cadre du chapitre (voir la section 3.4), nous avons comparé les résultats obtenus par les algorithmes NON-A avec ceux obtenus par un EKF. Nous avons alors montré que sous des hypothèses moins fortes, nous arrivions à obtenir des résultats équivalents voire meilleurs que pour l'EKF. Dans ce chapitre, nous effectuons uniquement les expérimentations sur les algorithmes NON-A. Pour les algorithmes NON-A-HM1-BIS, NON-A-HM2 et NON-A-HM3 les caractéristiques des erreurs sont sensiblement les mêmes en fonction des différents types de robot. Pour les algorithmes NON-A-HM4 et NON-A-HM5 nous avons encore une fois comme dans le cas des robots de type (2, 0), des erreurs relativement grandes (mais comparables à l'EKF dans le cas unicycle). Concernant les temps de calcul, il faut noter une augmentation de ce temps de calcul pour les algorithmes NON-A-HM3, NON-A-HM4 et NON-A-HM5 due à la fonction de calcul de l'intersection de deux cercles, relativement longue à l'exécution.

CONCLUSION DU CHAPITRE

Nous avons dans ce chapitre étendu les résultats obtenus pour un robot de type (2, 0) aux autres types de robots (notamment (3, 0), (2, 1) et (1, 1)). Nous utilisons pour cette généralisation la formulation dans le domaine complexe des modèles cinématiques. Cette formulation est rappelée pour chacun des types de robot dans la section 4.1. Cette écriture complexe présente des similitudes d'écriture entre les différents types de robots qui facilite donc l'extension des résultats obtenus dans le chapitre 3 aux autres types de robots.

Pour chacun des cas d'étude présentés dans la table 3.1 nous donnons donc un estimateur de posture et éventuellement de commandes pour chacun des quatre

TABLE 4.3 – Résultat de simulation pour un robot de type (2, 1)

Algorithmes	Résultats				
	erreur moyenne en distance [m]	variance moyenne de l'erreur en distance [m ²]	erreur moyenne sur v [m.s ⁻¹]	erreur moyenne sur ω [rad.s ⁻¹]	Temps d'exécution moyen [s]
<u>NON-A-HM1BIS</u>	0.2448	0.3379	0.0164	0.0009	0.1895
<u>NON-A-HM2</u>	0.1335	0.0025	0.0155	$4.9114 \cdot 10^{-6}$	0.0826
<u>NON-A-HM3</u>	0.1918	0.0021	0.0147	0.0928	0.6236
<u>NON-A-HM4</u>	2.2279	8.9471	X	X	0.6414
<u>NON-A-HM5</u>	2.1089	9.1358	X	X	0.6267

TABLE 4.4 – Résultat de simulation pour un robot de type (1, 1)

Algorithmes	Résultats				
	erreur moyenne en distance [m]	variance moyenne de l'erreur en distance [m ²]	erreur moyenne sur v [m.s ⁻¹]	erreur moyenne sur ω [rad.s ⁻¹]	Temps d'exécution moyen [s]
<u>NON-A-HM1BIS</u>	0.2412	0.0017	X	X	0.2124
<u>NON-A-HM2</u>	0.1839	0.0015	X	X	0.0813
<u>NON-A-HM3</u>	0.2173	0.0016	X	X	0.6381
<u>NON-A-HM4</u>	2.6381	3.2973	X	X	0.6810
<u>NON-A-HM5</u>	1.7060	1.9442	X	X	0.6822

types de robots étudiés. Quand cela est possible, nous utilisons les résultats développés dans le cadre du robot unicycle. Si ce n'est pas possible, la démonstration du résultat proposé est effectuée. Finalement une étude en simulation est effectuée pour valider les résultats obtenus de manière théorique. Nous montrons que les différents résultats sont quasiment équivalents entre les différents robots. Nous avons aussi montré dans le chapitre précédent le bon comportement de nos algorithmes dans le cadre du robot unicycle vis à vis de l'EKF. En conclusion les algorithmes proposés dans ce chapitre permettent de localiser la plupart des robots mobiles à roues et ce dans cinq cas d'étude différents dits unitaires (dans un cas avec plus de points d'intérêt une fusion des résultats obtenus dans les cas unitaires est tout à fait envisageable (voir la section 3.5)).

NAVIGATION AUTONOME D'UNE FLOTTILLE DE ROBOTS

SOMMAIRE

5.1	ALGORITHME DE PLANIFICATION DE TRAJECTOIRE	129
5.1.1	Rappel du modèle du robot	129
5.1.2	Planification de trajectoire : platitude et calcul variationnel . . .	130
5.1.3	Résolution et implémentation en temps réel	133
5.2	LE BLOC STRATÉGIE	133
5.2.1	Choix de la fonction coût	134
5.2.2	Choix de l'objectif	137
5.2.3	Modification des contraintes	140
5.2.4	Modification simultanée des contraintes et de la fonction coût .	141
5.2.5	Sur l'influence des différents paramètres	142
5.3	RÉSULTATS D'EXPÉRIMENTATIONS ET DE SIMULATIONS	143
5.3.1	Définition des critères de validation	144
5.3.2	Résultats d'expérimentations et de simulations	145
	CONCLUSION	149

LES systèmes multi-robots ont la possibilité de résoudre plus efficacement et plus rapidement une mission. Ainsi de nombreux domaines applicatifs sont naturellement concernés par de tels systèmes : exploration planétaire ou sous-marine, robotique de chantier (par exemple, le transport d'objets encombrants ou lourds à l'aide de plusieurs véhicules autonomes), surveillance de zones à risque (incendies, pollution). Pour cela, les robots doivent constituer un réseau au sein duquel ils peuvent échanger des informations afin d'accomplir une tâche complexe en commun. Notons, que l'on parle dans ce cas de robotique collaborative ou coopérative ; la distinction porte sur les objectifs de la flottille :

- accomplir les tâches attribuées (différentes) aux robots en tirant parti des informations échangées au sein de la flottille (par exemple afin d'éviter les collisions entre les robots) : on parle alors de collaboration,
- accomplir une tâche partagée (objectif commun pour toute la flottille) de façon plus efficace : on parle alors de coopération.

Notons que pour des applications de surveillance et de supervision ou d'exploration de zones il s'agira bien de coopération puisque la flottille partage un

même et unique objectif.

En introduction à ce mémoire, une architecture logicielle pour un robot appartenant à une flottille de robots qui doit naviguer de façon autonome dans un environnement partiellement connu a été présentée (voir figure 1.2). Nous rappelons ici cette figure :

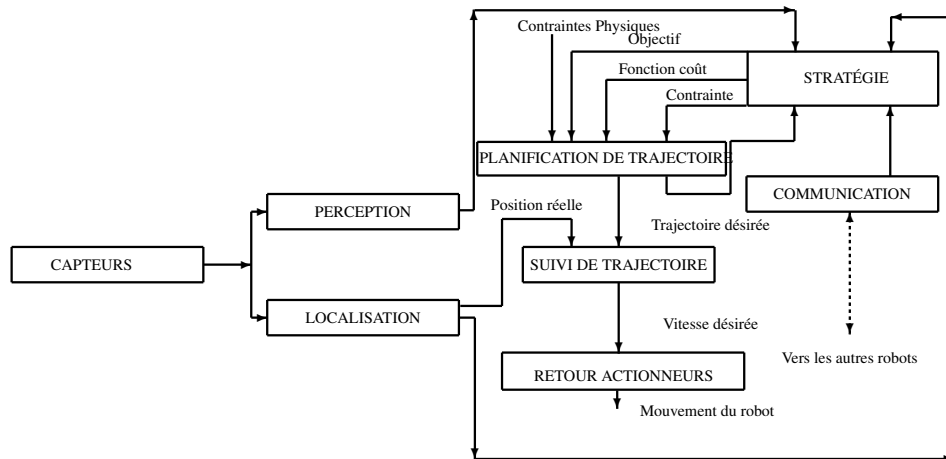


FIGURE 5.1 – Architecture logicielle d'un robot mobile à roues

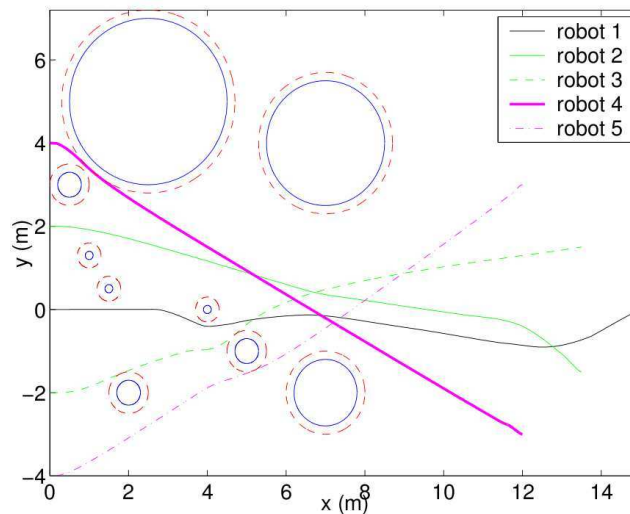
Dans ce chapitre, nous allons nous intéresser plus particulièrement au bloc de planification de trajectoire et au bloc de stratégie tout en gardant une vision décentralisée (en particulier il n'y a pas de planification globale pour la flottille). Le rôle du bloc de stratégie est de calculer une trajectoire optimale pour le robot en accord avec les informations en provenance du bloc de stratégie. Les contraintes cinématiques du robot ainsi que les contraintes de l'environnement sont prises en compte pour calculer une trajectoire optimale. L'algorithme proposé dans ce travail est une évolution de l'algorithme de planification de trajectoire pour une flottille décentralisée de robots, présenté dans [32, 33].

Nous développons ici un nouveau bloc, appelé bloc de stratégie, en interaction directe avec le bloc de planification de trajectoire. Grâce à ce bloc, il est possible de modifier en temps réel, pendant la tâche de navigation, l'objectif de la mission, les paramètres des contraintes (vitesse maximale, ...) ainsi que la fonction coût. Ces modifications se font en fonction des informations que le robot reçoit de ses propres capteurs ainsi que des robots voisins. L'intérêt de l'approche proposée est qu'elle permet de garder le caractère décentralisé de la flottille. En effet, chaque robot possédant son propre bloc stratégie il lui est donc possible d'adopter un comportement différent des robots de son voisinage. Il serait même possible d'envisager par exemple un robot « gentil » et un robot « méchant » par exemple (ce dernier cherchant à entrer en collision avec ses voisins). Le bloc de stratégie peut être vu comme l'intelligence du robot.

Les approches coopératives pour la navigation d'une flottille de robots sont, nous l'avons vu en introduction, classifiées en deux approches, centralisées (un seul calculateur se charge de déterminer pour chaque robot certains aspects haut niveau de la navigation) et décentralisées (on peut par exemple mentionner le cas d'une flottille avec une structure meneur/suiveur). Nous avons vu dans l'introduction que les approches centralisées facilitent l'implémentation sur des espaces très grands mais sont plus sensibles à une perte de communication entre le meneur et

ses suiveurs. L'objectif des approches décentralisées est de lever cette limitation tout en obtenant des résultats similaires à ceux obtenus pour une flottille centralisée.

Le présent travail s'appuie sur les travaux concernant la planification de trajectoire pour une flottille décentralisée de robots [32, 33], qui présente de bonnes performances en terme de temps de calcul, de volume d'informations échangées entre les robots, et de temps de navigation de la flottille. Une comparaison par rapport à l'état de l'art a été faite dans [32, 33] et est reprise dans la table 5.1. Cette table montre que l'algorithme utilisé pour la planification de trajectoire présente de meilleurs résultats que les autres algorithmes étudiés.



Approche	Centralisée	Meneur / Suiveur [76]	Faiblement décentralisée [72]	Complètement décentralisée [32, 33]
Temps maximum de résolution des conflits	2050ms	313ms	703ms	121ms
Type d'informations échangées	global	local	local	local
Implémentation	-- (impossible quand $N_a \gg 1$)	++ (résolution séquentielle)	- (difficile si conflit entre de nombreux robots)	+
Temps de navigation	35s	39s	36s	36.5s

TABLE 5.1 – Comparaison de l'algorithme complètement décentralisé [32, 33] avec d'autres algorithmes centralisés et décentralisés sur une flottille de 5 robots

Dans ce chapitre, les notations des chapitres précédents sont quelque peu adaptées pour prendre en compte l'aspect multi-robots. Ainsi,

- Chaque robot R_i de la flottille de N_a robots connaît son **état initial**, $\mathbf{p}_i(t_0) = (x_i \ y_i \ \theta_i)^T$ et son **état final désiré**, $\mathbf{p}_i(t_{final})$.
- Le point objectif du robot R_i est $G_i = [x_{G_i} \ y_{G_i}]$,
- La flottille évolue dans un environnement inconnu, chaque robot n'a que des informations locales à propos de son environnement (la taille du champ de perception dépend du type de capteurs utilisés). L'espace de l'ensemble des obstacles détectés dans le champ de perception du robot à l'instant t est noté $\mathcal{O}_i(t)$.
- Chaque robot de la flottille a un voisinage de robot(s) avec lequel il peut communiquer à l'instant t , ce voisinage est noté $W(R_i)(t)$.

Dans la section 5.1, l'algorithme de planification de trajectoire utilisé est présenté. L'un des principaux avantages de cette méthode, outre son aspect décentralisé, est la prise en compte du modèle cinématique du robot et des différentes contraintes du problème (non collision entre les robots, entre les robots et les obstacles, ...). Elle s'appuie sur la structure algébrique du robot mobile. En effet grâce à la propriété de platitude, la planification de trajectoire peut s'atteler uniquement à l'estimation des sorties plates, les autres variables du système étant fonction de ces sorties plates et de leurs dérivées. La prise en compte des contraintes (à condition qu'elles s'écrivent en fonction des sorties plates puisque ce sont les seules variables optimisées) se fait en utilisant un algorithme d'optimisation non linéaire sous contraintes. Il faut donc choisir les contraintes qui seront intégrées au problème d'optimisation ainsi que le critère d'optimisation (la fonction coût). Dans la section 5.2 un algorithme permettant justement de choisir la fonction coût (comme une somme pondérée de cinq fonctions coût dites unitaires) est proposé. La démarche à suivre pour modifier les contraintes et l'objectif en temps réel est aussi présentée. Des exemples simples permettant de bien comprendre les effets de ces modifications sont présentés en guise d'illustrations. Dans la section 5.3 des résultats d'expériences et de simulations permettant de valider à travers l'étude de quatre critères l'algorithme proposé sont présentés. Ainsi, l'extensibilité, la complexité, le choix des paramètres ainsi que l'adaptabilité sont étudiés.

5.1 ALGORITHME DE PLANIFICATION DE TRAJECTOIRE

Nous avons dans ce mémoire décidé d'utiliser une approche décentralisée combinant les intérêts des solutions discrètes et continues. Outre la capacité des approches décentralisées à proposer une solution là où l'approche centralisée est défaillante (notamment si le meneur ne peut plus transmettre aux suiveurs les informations nécessaires à leur navigation) l'approche discrète/continue nous permet de tirer bénéfice de travaux effectués tant du point de vue intelligence artificielle que du point de vue contrôle des systèmes. Le principal défaut des approches discrètes est leur incapacité à proposer une solution continue sur l'ensemble de l'espace en gardant une complexité combinatoire raisonnable. Les approches continues donnent de bonnes solutions d'un point de vue local, mais présentent des difficultés lorsqu'il s'agit de planifier une trajectoire sur un environnement de grande surface. Une illustration de ces propos est celle de la planification de trajectoire pour une voiture. Lorsqu'un GPS routier calcule un itinéraire, le trajet proposé est un trajet discret ; on planifie les points de passage, les grandes villes étapes, mais cela ne dispense pas de conduire le véhicule localement en fonction des aléas de la route : bouchon, déviation, changement de voies Il en va de même pour le robot, il faut planifier des trajectoires haut niveau permettant d'explorer largement l'environnement pour satisfaire la mission tout en utilisant une planification «locale» pour les morceaux de trajectoire. C'est cette planification locale que nous allons étudier dans cette section.

5.1.1 Rappel du modèle du robot

Nous considérons dans ce chapitre un robot unicycle (c'est-à-dire un robot de type $(2, 0)$ selon la classification présentée dans la section 2.1.4.6). Un tel robot est décrit par un vecteur d'état de dimension trois qui est composé de sa position (x, y) et de son orientation θ . Le point de coordonnées (x, y) est le point milieu de l'axe reliant les deux roues motrices, l'orientation θ correspond à l'orientation du robot dans un repère préalablement fixé. Le modèle cinématique d'un tel robot est le suivant :

$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{p}, \mathbf{u}) = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{bmatrix}, \quad (5.1)$$

où $\mathbf{p} = (x, y, \theta)^T$ est la posture du robot (dans \mathbb{R}), et $\mathbf{u} = (v, \omega)^T$ est le vecteur de contrôle (vitesse linéaire v et vitesse angulaire ω).

En utilisant le changement de variable suivant,

$$z = x + iy \quad (5.2)$$

$$\Theta = \exp(i\theta) \quad (5.3)$$

le modèle cinématique du robot de type $(2, 0)$ dans $\mathbb{C}[X]$ s'écrit :

$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{p}, \mathbf{u}) = \begin{bmatrix} \dot{z} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} v\Theta \\ i\omega\Theta \end{bmatrix} \quad (5.4)$$

avec $\mathbf{p} = (z, \Theta)^T$, $\mathbf{p} \in \mathbb{C}^2$ la posture du robot, et $\mathbf{u} = (v, \omega)^T$ le vecteur de contrôle. Notons que l'application permettant de passer du modèle décrit dans le domaine réel à celui dans le domaine complexe est complètement bijective.

Pour planifier une trajectoire, nous utilisons la propriété de platitude du système. Il est donc possible de définir les sorties plates du système (ici z) et d'exprimer toutes les autres variables du système en fonction de ces sorties plates et de leur dérivées :

$$z = z, \quad (5.5)$$

$$\Theta = \pm \frac{\dot{z}}{|\dot{z}|}, \quad (5.6)$$

$$v = \pm |\dot{z}|, \quad (5.7)$$

$$\omega = \mp i \frac{\dot{\Theta}}{\Theta}, \quad (5.8)$$

nous rappelons que les indéterminations sur les signes se lèvent sans difficulté et que par la suite \pm et \mp seront omis pour faciliter la lecture. En optimisant uniquement la sortie plate z , il est possible de reconstruire le vecteur de posture, et le vecteur de commande qui correspondra alors à la trajectoire optimisée. Le problème de planification de trajectoire devient alors un simple problème de calcul variationnel n'ayant qu'une seule variable de décision z .

5.1.2 Planification de trajectoire : platitude et calcul variationnel

Nos objectifs sont que les robots puissent naviguer :

1. sans collision dans un environnement inconnu avec obstacles,
2. en respectant des contraintes physiques (vitesses et accélérations maximales, ...),
3. en respectant des contraintes géométriques de formation,
4. en respectant des contraintes de communication entre les robots au sein de la formation.

Ces objectifs conduisent à envisager un algorithme de planification de trajectoire pour un système non holonome basé sur une optimisation sous contraintes. En effet, la formulation du problème de planification comme un problème de commande optimale permet une flexibilité de l'algorithme afin de pouvoir d'une part être adapté au cadre multi-robots et d'autre part tenir compte de multiples contraintes.

Notons qu'une carte complète de l'environnement n'est généralement pas connue par avance puisque les obstacles sont détectés au fur et à mesure du déplacement du robot (en fonction de son rayon de perception). Ainsi, seule une certaine zone autour du robot est prise en compte. Dans cette région, nous supposons que l'environnement est parfaitement connu. Par conséquent, puisque l'environnement est exploré graduellement, une trajectoire joignant la configuration initiale à la configuration finale doit être calculée graduellement au cours du temps tant que la mission n'est pas achevée. Il est donc nécessaire de faire appel à une stratégie de planification sur un horizon glissant. Sur chaque horizon, une partie de la trajectoire est planifiée en ligne, en résolvant un problème de commande optimale sous contraintes [32].

Ainsi, il nous faut dans un premier temps préciser les notions de chemins et de trajectoires, puis donner une formulation mathématique de notre problème d'optimisation et enfin rappeler les principaux résultats développés dans la thèse de Michael Defoort [33].

5.1.2.1 Chemins et trajectoires

La plupart des travaux de planification de trajectoire sont basés sur le concept d'espace des configurations du robot introduit dans [91] au début des années 80. Une **configuration** désigne l'ensemble des paramètres caractérisant d'une manière unique le robot dans son environnement ou espace de travail. L'ensemble des configurations du robot est l'**espace des configurations** \mathcal{Q} , qui a une structure de variété différentielle. On note n la dimension de \mathcal{Q} .

Par exemple pour un robot de type unicycle (deux roues fixes centrées sur un même axe et une roue orientable décentrée), la configuration est le vecteur de pose augmenté des angles décrivant les rotations de chacune des roues.

Une **trajectoire** est une fonction continue de $[t_{initial}, t_{final}] \subset \mathbb{R}$ dans \mathcal{Q} qui à toute valeur $t \in [t_{initial}, t_{final}]$ associe une configuration :

$$\begin{aligned} q : [t_{initial}, t_{final}] &\rightarrow \mathcal{Q} \\ t &\mapsto q(t) \end{aligned}$$

Une trajectoire est dite **admissible** si elle est solution du système d'équations différentielles correspondant au modèle cinématique du robot, incluant les contraintes sur les commandes, pour des conditions initiales et finales données. Un **chemin** est l'image d'une trajectoire dans \mathcal{Q} . Un **chemin admissible** est l'image d'une trajectoire admissible.

5.1.2.2 Planification de trajectoire

Une partie essentielle de l'autonomie des véhicules réside en la capacité à planifier des trajectoires admissibles assurant des déplacements sans collision dans un environnement particulier. Cet environnement contient généralement des zones dans lesquelles le robot ne peut pas se déplacer. Ces zones peuvent être détectées lorsque le robot se déplace. Un cas particulier d'évitement d'obstacles non stationnaires est l'évitement de collisions avec d'autres robots. Le robot doit ainsi avoir la capacité de calculer ou de mettre à jour sa trajectoire en temps réel tant que la mission n'est pas achevée. Ainsi, dans notre cas, le problème de planification de trajectoire pour un robot mobile peut se définir de la manière suivante.

Définition 5.1 *On appelle **planification de mouvement de configuration**, le calcul d'une trajectoire admissible et sans collision pour un robot entre une configuration de départ et une configuration d'arrivée données.*

Il s'agit bien de courbes paramétrées par le temps.

Parfois, il est plus intéressant d'un point de vue pratique de s'intéresser non pas à la configuration d'un robot mais plus à sa posture, ce qui nous conduit à la définition suivante :

Définition 5.2 *On appelle **planification de mouvement de posture**, le calcul d'une trajectoire admissible et sans collision pour un robot entre une posture de départ et une posture d'arrivée données.*

Dans la littérature lorsqu'on parle de **planification de trajectoire** il s'agit en fait de planification de mouvement de posture.

5.1.2.3 Formulation mathématique

Le problème de planification consiste à calculer pour le robot d'indice i , la trajectoire admissible et sans collision, joignant la pose initiale $\mathbf{p}_i(t_{initial}) = \mathbf{p}_{i,initial}$ à la pose finale $\mathbf{p}_i(t_{final}) = \mathbf{p}_{i,final}$ (avec des commandes (ce sont les vitesses) initiales $\mathbf{u}_i(t_{initial}) = \mathbf{u}_{i,initial}$ et finales $\mathbf{u}_i(t_{final}) = \mathbf{u}_{i,final}$ qui peuvent être fixées ou laissées libres) et optimisant un critère défini. La fonction critère peut être une mesure du temps, d'énergie ou un critère plus complexe tel que la visibilité. Formulé ainsi, il s'agit d'un problème de contrôle optimal sous contraintes [33] : Il faut donc trouver le vecteur de commande $\mathbf{u}_i(t)$ et la trajectoire optimale de chacun des robots qui minimise la fonction coût (5.9) :

$$J = \int_{t_0}^{t_f} L_{(R_i)}(z_i, \dot{z}_i, \mathbf{u}_i, t) dt, i \in \{1, \dots, N_a\} \quad (5.9)$$

où $L_{(R_i)}(z_i, \dot{z}_i, \mathbf{u}_i, t)$ est la fonction coût qui peut être choisie dans une banque de fonctions. En utilisant la propriété de platitude on peut alors exprimer \mathbf{u}_i comme une fonction des sorties plates et de leurs dérivées (en nombre fini). Pour les robots unicycle et voiture on aura donc : $\mathbf{u}_i = f(z_i, \dot{z}_i, \dots, z_i^{(\ell)})$ avec $\ell = 2$ pour les robots de type unicycle et 3 pour les robots de type voiture. Tous les robots mobiles non dégénérés sont plats (cf. Chapitre 2), ainsi les développements ci-dessous seront facilement transposables aux autres cas ; grâce à la platitude, la trajectoire optimale en z_i permettra d'en déduire la commande optimale à appliquer et ce couple sera ainsi solution de l'équation (5.1), qui est le modèle cinématique du robot R_j . La fonction coût, ainsi obtenue, admet comme unique variable de décision z_i et elle doit être minimisée en satisfaisant les contraintes suivantes : $\forall t \in [t_0, t_f]$:

- C_1 La commande est bornée : $|\mathbf{u}_i| \leq \mathbf{u}_{max}$,
- C_2 Le robot doit satisfaire les conditions initiale et terminale : $\mathbf{p}_i(t_{initial}) = \mathbf{p}_{i,initial}, \mathbf{p}_i(t_{final}) = \mathbf{p}_{i,final}$,
- C_3 Le robot doit garder une distance minimale d'au moins $d(R_i, O_{mi})_{min}$ entre lui et tous les obstacles détectés O_{mi} , appartenant à un ensemble $\mathcal{O}_i(t)$ de tous les obstacles détectés à un instant t , c'est-à-dire que $|z_{O_i} - z_{R_i}| \geq d(R_i, O_{mi})_{min}$,
- C_4 Le robot doit garder une distance minimale d'au moins $d(R_i, R_j)_{min}$ entre lui et tous les robots j , appartenant à son voisinage à l'instant $W(R_i)(t)$, c'est-à-dire que $|z_{R_i} - z_{R_j}| \geq d(R_i, R_j)_{min}$.

Notons que ces contraintes sont ainsi des fonctions des sorties plates et de leurs dérivées (en nombre fini) : $C_k = f_k(z_i, \dot{z}_i, \dots, z_i^{(\ell)})$, $\forall k \in \llbracket 1 \ 4 \rrbracket$. Par ailleurs cette démarche permet d'intégrer d'autres contraintes physiques du moment qu'elles sont fonction des sorties plates : par exemple on peut tenir compte de distances maximales entre certains robots pour préserver les liens de communications existant entre ces robots. Ainsi toutes les contraintes C_k , $\forall k \in \llbracket 1 \ 4 \rrbracket$ peuvent s'exprimer comme une fonction de z_i et d'un nombre fini de ses dérivées successives (2 si l'on se restreint aux contraintes présentées ici pour un robot de type unicycle).

Remarque 5.1 *Étant donnée la notion de voisinage, le robot ne peut communiquer à chaque instant qu'avec les robots appartenant à ce voisinage. Il est donc nécessaire d'avoir un processus permettant de savoir quel robot appartient au voisinage d'un robot à un instant donné.*

5.1.3 Résolution et implémentation en temps réel

Nous avons vu qu'en raison des propriétés de platitude, la fonction coût (équation (5.9) dans notre cas) et les contraintes (C_1-C_4 dans notre cas) peuvent s'exprimer en fonction des sorties plates et d'un nombre fini de leurs dérivées successives jusqu'à un ordre donné (2 dans notre cas). Ainsi, la solution du problème optimal dépend uniquement de la sortie plate et de ses dérivées [46].

Le problème de contrôle optimal est donc transformé en un problème d'optimisation paramétrique en utilisant des polynômes de degré 3 (ou des B-splines comme dans [33]) pour décrire la sortie plate du robot (le choix de l'ordre du polynôme utilisé est à la discrétion de l'utilisateur : il faut choisir un ordre suffisamment élevé pour présenter une bonne variété de trajectoires possibles sans rendre le problème trop complexe d'un point de vue calculatoire). Comme le robot ne connaît pas à l'avance l'environnement dans lequel il évolue, il n'est pas possible de calculer les trajectoires de tous les robots en une seule fois. Il faut donc calculer ces trajectoires sur un horizon temporel glissant. La trajectoire de chaque robot de la flottille est donc calculée sur un intervalle de temps $[\tau_k, \tau_k + T_c]$, où T_c est la période de mise à jour de la trajectoire. Cette trajectoire est calculée pour être valide sur un horizon T_p avec $T_p \gg T_c$, τ_k la période d'échantillonnage. Le problème d'optimisation

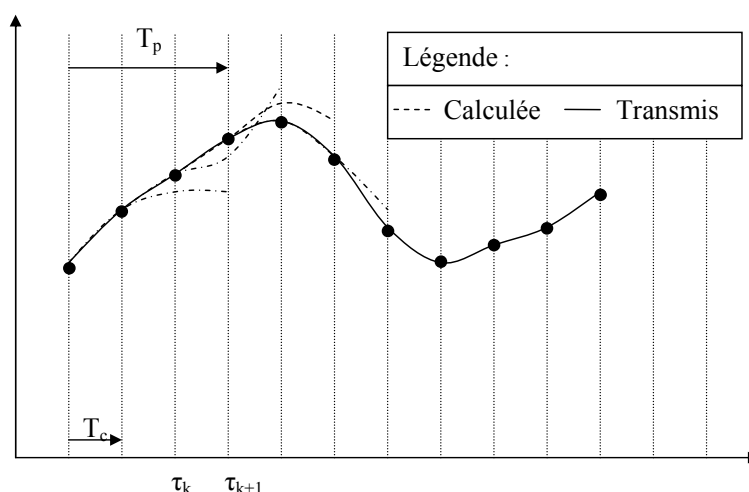


FIGURE 5.2 – Horizons de calculs et de planification.

est alors résolu en utilisant le solveur CFSQP [79] (n'importe quel solveur peut être utilisé, par exemple [56] utilise une approche par intervalle pour résoudre ce problème).

Nous allons montrer dans le prochain paragraphe que l'implémentation de différents *scenarii* est facilitée en utilisant cette approche. En effet il est possible de :

- changer la fonction coût au cours du temps voire même de faire une fusion de plusieurs fonctions coût,
- de modifier les contraintes pendant la tâche de navigation,
- de coopérer avec les voisins pour faciliter la tâche de navigation.

5.2 LE BLOC STRATÉGIE

Nous avons vu en introduction que ce bloc est «l'intelligence» du robot. C'est lui qui permet, en lien avec le bloc de planification de trajectoire, de tirer bénéfice

de l'utilisation conjointe des approches continue et discrète. Les différentes entrées de ce bloc sont les suivantes :

- Les informations en provenance des voisins,
- Les informations en provenance de l'opérateur,
- Les informations en provenance des blocs de localisation et perception.

Les différentes sorties sont les suivantes

- Le prochain point de passage (en fonction d'un algorithme de planification haut-niveau) . . . ,
- Les contraintes (qui doivent s'exprimer comme une fonction différentielle des sorties plates), ainsi que les paramètres de ces contraintes. Il est par exemple possible d'augmenter ou de diminuer la vitesse maximale admissible en fonction des limitations de vitesse de l'environnement, ou alors d'augmenter ou de diminuer les distances de sécurité en fonction de la vitesse actuelle . . . ,
- La fonction coût (qui doit s'exprimer comme une fonction différentielle des sorties plates), par exemple, le robot peut décider de minimiser l'énergie consommée, ou au contraire décider de terminer le plus rapidement sa mission, . . .

Ces différentes sorties peuvent varier à chaque période d'échantillonnage et ce en fonction des informations disponibles en entrées. Dans la suite de cette section, une proposition d'une fonction coût, fusion de cinq autres fonctions coût unitaires sera donnée. Ensuite, un algorithme permettant de choisir un objectif intermédiaire par la coopération avec les voisins sera présenté. Finalement, il sera montré l'influence du choix des paramètres des contraintes sur la trajectoire du robot. Un exemple illustratif sera donné dans chaque situation pour faciliter la compréhension des phénomènes.

5.2.1 Choix de la fonction coût

Il est possible durant la tâche de navigation, à chaque période de mise à jour T_c , de changer la fonction coût. Cette modification de la fonction coût a un impact direct sur la trajectoire empruntée par le robot ; par exemple, si c'est l'énergie qui est minimisée, le robot va réduire sa vitesse par rapport à une trajectoire où il cherche à atteindre son objectif le plus rapidement possible. De même si les trajectoires présentant un coefficient de sécurité plus important sont favorisées, alors le robot prendra plus de marge de sécurité par rapport à ses voisins ou par rapport aux obstacles. Une fonction coût, issue d'une somme pondérée de cinq fonctions coût unitaires, est donnée dans la suite de cette section.

5.2.1.1 Banque de fonctions coût

Cinq fonctions coût dites unitaires sont introduites dans cette sous-section. Notons que de nombreuses autres fonctions coût unitaires peuvent être utilisées à condition qu'elles s'expriment sous la forme d'une fonction différentielle des sorties plates. Pour chaque robot i on note G_i la position finale désirée.

- L_1 : \mathcal{L}_1 —norme du vecteur $\overrightarrow{R_i G_i}$,
- L_2 : \mathcal{L}_2 —norme du vecteur $\overrightarrow{R_i G_i}$,
- L_3 : consommation énergétique,
- L_4 : l'inverse de la distance entre le robot R_i et les obstacles O_i , $\forall O_i \in \mathcal{O}_i(t)$,

	Expression mathématique
L_1	$ x_{G_i} - x_{R_i} + y_{G_i} - y_{R_i} $
L_2	$\sqrt{(x_{G_i} - x_{R_i})^2 + (y_{G_i} - y_{R_i})^2}$
L_3	v^2
L_4	$\frac{1}{\sqrt{(x_{O_i} - x_{R_i})^2 + (y_{O_i} - y_{R_i})^2}}$
L_5	$\frac{1}{\sqrt{(x_{R_j} - x_{R_i})^2 + (y_{R_j} - y_{R_i})^2}}$

TABLE 5.2 – Expression mathématique des fonctions coût

– L_5 : l'inverse de la distance entre le robot R_i et le R_j , $\forall R_j \in W(R_i)(t)$.

La table 5.2 donne les expressions mathématiques des cinq fonctions coût présentées. Il est possible d'utiliser chacune de ces fonctions coût indépendamment les unes des autres. Cependant, et c'est ce qui sera proposé dans la suite, il est intéressant d'utiliser une combinaison de ces différentes fonctions coût pour obtenir des comportements plus complexes.

5.2.1.2 Détermination de la fonction coût par fusion

Dans la sous-section précédente, cinq fonctions coût unitaires permettant de réduire la distance (au sens \mathcal{L}_1 ou \mathcal{L}_2), la consommation énergétique, ou permettant de maximiser la distance aux obstacles ou aux autres robots, ont été présentées. À défaut d'utiliser chacune d'entre elles indépendamment les unes des autres, il est intéressant d'effectuer une fusion des différentes fonctions coût pour en produire une permettant d'obtenir des comportements plus complexes. Cette fusion est effectuée de la façon suivante :

$$L_{(R_i)} = \sum_{k=1}^5 \left(\left(\prod_{l=1, l \neq k}^5 (1 - \mu_l(t)) \right) \mu_k(t) L_k \right), \quad (5.10)$$

$$\sum_{k=1}^5 \mu_k(t) = 1. \quad (5.11)$$

Où :

- $\mu_1(t) \in [0 1]$ est le poids de la \mathcal{L}_1 –norme de la distance à l'objectif,
- $\mu_2(t) \in [0 1]$ est le poids de la \mathcal{L}_2 –norme de la distance à l'objectif,
- $\mu_3(t) \in [0 1]$ est l'inverse de la charge de la batterie du robot, c'est-à-dire qu'il vaut 1 si le robot est complètement déchargé et 0 si le robot est complètement chargé,
- $\mu_4(t) \in [0 1]$ est un paramètre destiné à sécuriser la non-collision entre le robot et les obstacles (la non-collision avec les obstacles est déjà une contrainte mais l'algorithme d'optimisation ira à la limite de la contrainte si cela facilite la minimisation de la fonction coût. En ajoutant ce terme, il est possible de privilégier les trajectoires qui n'iront pas à la limite de collision),
- $\mu_5(t) \in [0 1]$ est un paramètre destiné à sécuriser la non-collision entre les robots (la non-collision entre les robots est déjà une contrainte mais l'algorithme d'optimisation ira à la limite de la contrainte si cela facilite la minimi-

sation de la fonction coût. En ajoutant ce terme, il est possible de privilégier les trajectoires qui n'iront pas à la limite de collision),

La fusion proposée donne une fonction coût globale dépendante du temps. En effet, le paramètre $\mu_3(t)$ va diminuer au cours du temps en fonction de l'utilisation du robot. $\mu_1(t)$ et $\mu_2(t)$ sont fonctions de la forme de la trajectoire désirée et peuvent être adaptés au cours du temps (ce paramètre reflète l'état de charge de la batterie du robot). Finalement $\mu_4(t)$ et $\mu_5(t)$ peuvent être changés si au cours de la mission il est nécessaire que le robot adopte un comportement plus sûr en fonction de contraintes environnementales externes.

Une fonction coût fusion de plusieurs autres coûts unitaires a été proposée. La sous-section suivante donne des critères de choix des différents paramètres entrant en compte dans la planification de trajectoire.

5.2.1.3 Influence des poids sur la planification

Afin d'étudier l'influence des poids sur la forme de la trajectoire planifiée, nous présentons deux *scenarii* de simulation. Le premier scénario (voir la figure 5.3(a)) est un scénario de croisement entre deux robots sans obstacles. Chacun des robots doit se diriger vers le point diagonalement opposé à son point de départ, ce qui entraîne donc un croisement entre les deux robots. L'objectif est de démontrer l'influence du paramètre $\mu_5(t)$ sur la trajectoire des robots. Dans le premier cas (voir la figure 5.3(b)) la seule fonction coût minimisée est la \mathcal{L}_2 -norme de la distance à l'objectif. Bien qu'il n'y ait pas de collision entre les robots (c'est une contrainte), les distances de sécurité ne sont pas très importantes. Pour les deux expériences suivantes (figure 5.3(c) et figure 5.3(d)) nous ajoutons l'influence de la fonction coût de non-collision entre robots en plus de la minimisation de la \mathcal{L}_2 -norme de la distance à l'objectif. La trajectoire empruntée par les robots est clairement déviée de manière à éviter au maximum les risques de collisions. Le dernier cas s'approche presque de la situation où il est plus important d'éviter son voisin que d'atteindre son voisinage ce qui pourrait être une cause de blocage. Il faut donc bien régler les paramètres pour obtenir le comportement désiré.

Le deuxième scénario (voir la figure 5.4(a)) est cette fois encore un scénario de croisement entre les robots où chacun doit se diriger vers le point diagonalement opposé à son point de départ ce qui entraîne donc un croisement entre les deux robots. Cependant, à la différence du premier scénario, un obstacle a été ajouté sur le point de croisement. L'objectif est de démontrer l'influence du paramètre $\mu_4(t)$ sur la trajectoire des robots. Dans le premier cas (voir la figure 5.4(b)) la seule fonction coût minimisée est la \mathcal{L}_2 norme de la distance à l'objectif. Bien qu'il n'y ait pas de collision entre les robots et l'obstacle (c'est une contrainte), les distances de sécurité ne sont pas très importantes. Pour les deux expériences suivantes (figure 5.4(c) et figure 5.4(d)) nous ajoutons l'influence de la fonction coût de non-collision entre le robot et les obstacles en plus de la minimisation de la \mathcal{L}_2 -norme de la distance à l'objectif. La trajectoire empruntée par les robots est déviée de manière plus importante lorsque la valeur de μ_4 augmente afin d'éviter au maximum les risques de collisions.

Remarque 5.2 Dans les deux scénarios, $\mu_1(t) = \mu_3(t) = 0$. $\mu_4(t)$ et $\mu_5(t)$ sont nulles alternativement.

Dans les deux cas, l'influence des paramètres de sécurité est bien visible

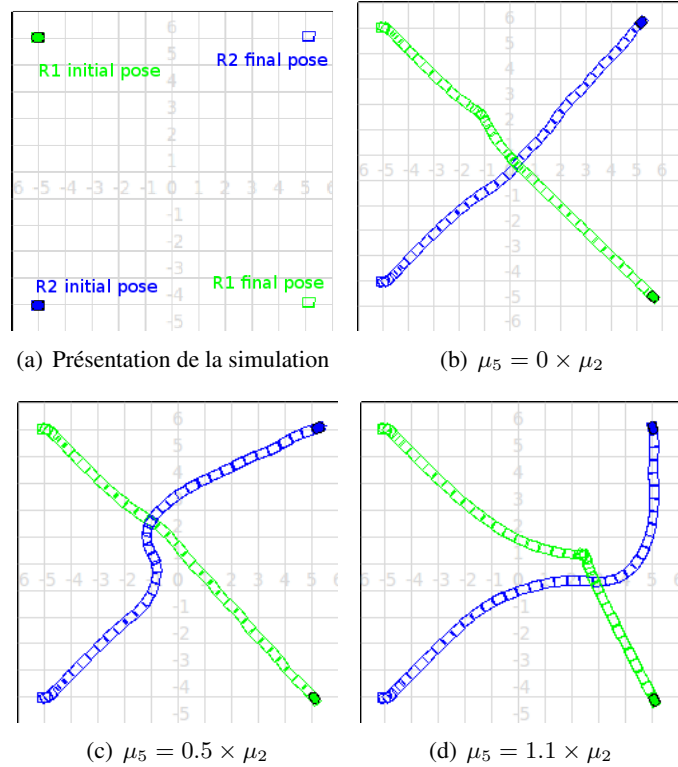


FIGURE 5.3 – *Modification de la fonction coût - effet sur la trajectoire*

puisque l'on passe d'une situation où les contraintes sont tout juste respectées (à leurs limites de rupture) à une situation où les contraintes sont très largement respectées. Il est donc possible en ajustant les paramètres d'obtenir des comportements de robots différents. Il faut donc les adapter pour obtenir le comportement désiré.

5.2.2 Choix de l'objectif

Dans la section précédente, une fonction coût dont le choix dépend d'une somme pondérée de cinq fonctions coût élémentaires a été proposée. Elle permet une navigation sûre et autonome du robot. Cependant, la fonction coût définie par l'équation (5.9) est indépendante des informations en provenance du voisinage du robot. Sous certaines conditions, le robot peut décider de coopérer avec son voisinage de façon à mener à bien sa propre mission. Pour cela, à chaque instant, le robot calcule une fonction de satisfaction notée \mathcal{F}_{ii} , qui dépend de la distance entre le robot et son objectif et du taux d'occupation de l'espace autour de lui. Il calcule selon le même processus la fonction de satisfaction \mathcal{F}_{ij} des robots de son voisinage (en utilisant les informations données par ses voisins). Avec toutes ces informations, un algorithme de coopération est utilisé pour décider si le robot doit ou non coopérer avec ses voisins et choisir avec lequel.

5.2.2.1 Calcul de la fonction de satisfaction

La fonction de satisfaction \mathcal{F}_{ij} donne la satisfaction du robot j dans le référentiel du robot i (ce qui permet de comparer la satisfaction propre du robot i par rapport à celle qu'il aurait s'il était à la place du robot j). Cette notion de référentiel est très importante. En effet, pour évaluer l'intérêt que le robot i a à coopérer

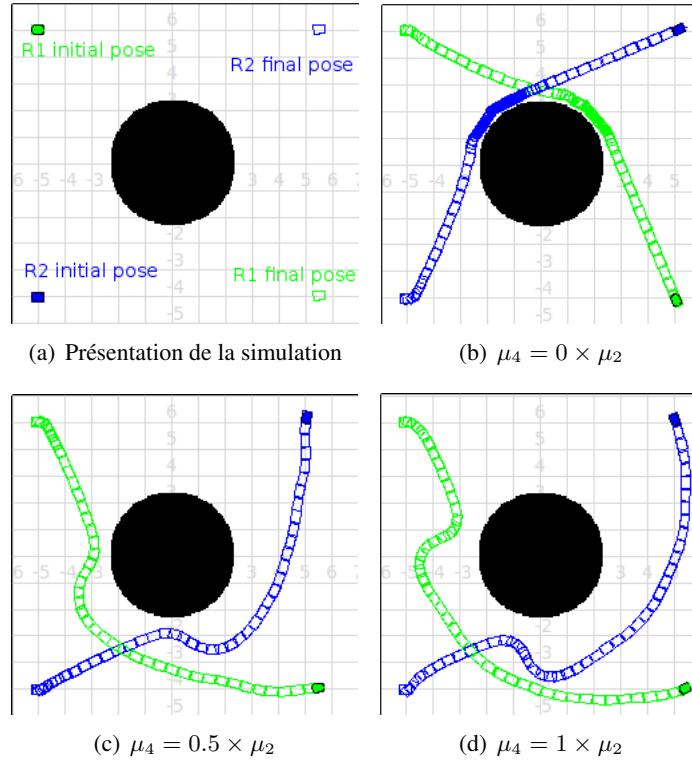


FIGURE 5.4 – Modification de la fonction coût - effet sur la trajectoire

avec le robot j , il faut qu'il « imagine » être à la place du robot j et qu'il évalue sa fonction de satisfaction. Il peut alors effectuer une comparaison avec la fonction de satisfaction qu'il a à sa propre place.

La fonction satisfaction dépend de deux fonctions :

1. $f_j^C(t)$: La fonction de congestion, qui donne l'état d'occupation de l'espace autour du robot R_j . Plus le taux d'espace libre autour du robot est important et plus la navigation sera aisée,
2. $f_{ij}^F(t)$: La fonction d'accomplissement de la tâche, qui donne la distance entre le robot R_j et l'objectif du robot R_i .

La fonction de congestion du robot R_i est calculée, en utilisant les notations de la figure 5.5, de la façon suivante :

$$f_i^C(t) = \frac{\text{Surface libre}_i(t)}{\text{Surface de perception}_i(t)}. \quad (5.12)$$

La fonction d'accomplissement de la tâche du robot R_j dans le référentiel du robot R_i est calculée de la façon suivante :

$$f_{ij}^F(t) = \max \left(0; 1 - \frac{d_{R_j, \text{objectif}_{R_i}}(t)}{d_{R_i, \text{objectif}_{R_i}}(t=0)} \right). \quad (5.13)$$

Remarque 5.3 La fonction \max permet d'éviter des valeurs plus petites que zéro, qui ne seraient pas homogènes avec les valeurs des autres fonctions. Ce problème pourrait survenir si, durant le mouvement, le robot devait se rendre dans un endroit plus éloigné de son point d'arrivée qu'il ne l'était au point de départ (par exemple pour contourner un obstacle).

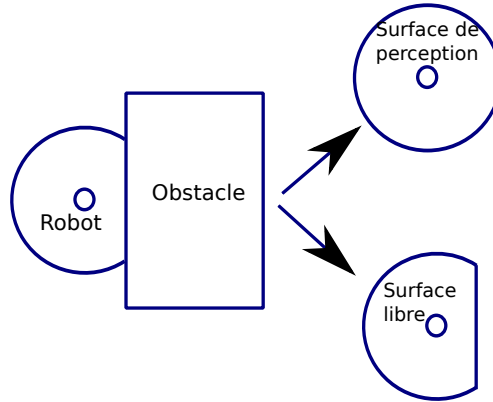


FIGURE 5.5 – Surface pour le calcul de la fonction de congestion

Finalement la fonction de satisfaction globale du robot R_j dans le référentiel du robot R_i est calculée de la façon barycentrique suivante :

$$\mathcal{F}_{ij}(t) = \frac{f_i^C(t) + \beta f_{ij}^F}{\beta + 1}, \beta \in \mathbb{R}^+. \quad (5.14)$$

Le paramètre β permet de donner l'influence de la fonction d'accomplissement de la tâche par rapport à la fonction de congestion de l'espace. Dans la suite du chapitre, une étude de l'influence de ce paramètre sera faite (notons qu'il est forcément positif ou nul).

5.2.2.2 Algorithme de coopération

Une fois la fonction de satisfaction calculée, il est nécessaire de mettre en place un algorithme de coopération entre le robot R_i et ses voisins $R_j \in W(R_i)(t)$ en prenant comme entrées les valeurs numériques des fonctions \mathcal{F}_{ij} . La table 5.3 donne l'algorithme de coopération proposé.

Pour chaque voisin, la fonction de satisfaction $\mathcal{F}_{ij}(t)$ est calculée. Si cette fonction est α plus grande que la fonction de satisfaction du robot R_i notée $\mathcal{F}_{ii}(t)$, le voisin est ajouté dans l'ensemble des robots coopérants noté $\mathcal{C}(R_i)$. C'est à dire que le robot R_i aurait un gain de satisfaction de α fois en étant à la place du robot R_j . Si le paramètre $\alpha \in [0, +\infty[$ est égal à 0 alors tous les voisins vont être ajoutés à l'ensemble de coopération même si ledit voisin n'est pas intéressant. Si α est égal à 1, tous les voisins qui donneraient une satisfaction égale à celle du robot lui-même vont être ajoutés à l'ensemble de coopération. Finalement, pour $\alpha > 1$, plus α sera grand et moins le robot va ajouter de voisins à son ensemble de coopération.

Une fois l'ensemble de coopération construit, le robot R_i va prendre pour objectif intermédiaire la dernière position connue du robot appartenant à l'ensemble de coopération avec le plus grand \mathcal{F}_{ij} .

Le paramètre β est utilisé pour pondérer la fonction d'accomplissement de la tâche par rapport à la fonction de congestion de l'espace. Plus β est grand, et plus les robots voisins, proches de l'objectif final du robot étudié, vont être choisis pour la coopération. Si β est petit, les voisins disposant d'un plus grand espace libre de navigation que les autres seront choisis pour la coopération.

La figure 5.6 présente l'impact du choix de α et β sur la trajectoire du robot. Le robot R_2 doit faire un choix : soit il va vers le haut, soit il va vers le bas. Sans

coopération il fera un choix aléatoire (voir la figure 5.6(a)). Avec la coopération, le comportement dépendra du choix des paramètres α et β .

Deux sous-cas sont présentés : le premier, visible sur les figures 5.6(b) et 5.6(c), où chaque robot cherche à maximiser l'espace libre pour naviguer ($\beta < 1$). Dans le second cas, visible sur les figures 5.6(d) et 5.6(e) où la fonction de congestion de l'espace et celle d'accomplissement de la tâche ont le même poids. Pour ces deux *scenarii*, le comportement de coopération est plus important dans le premier cas (figures 5.6(b) et 5.6(d)) que dans le deuxième (figures 5.6(c) et 5.6(e)).

Dans la première figure sans coopération, le robot R_2 fait le choix (aléatoire) d'aller vers le haut. Pour un observateur externe, qui a la connaissance complète de la carte, il est évident que ce choix n'est pas le bon. Dans la deuxième figure, la coopération entre robots est assez importante et cette coopération se base uniquement sur l'augmentation de l'espace libre pour naviguer. Cela entraîne donc le robot R_1 vers le bas pour qu'il puisse obtenir plus d'espace de navigation. Dans la troisième figure où le comportement de coopération est diminué par rapport au cas précédent, le robot R_1 estime que le gain d'espace n'est pas suffisant pour s'éloigner de sa position courante. Dans les deux derniers cas, les comportements sont sensiblement les mêmes. Le robot R_3 étant toujours un petit peu en avance sur les autres, le robot R_2 et le robot R_1 coopèrent longtemps avec lui car il reste toujours le plus proche de l'objectif final de ses voisins.

Algorithme pour la coopération

```

Calculer  $\mathcal{F}_{ii}$  (équations (5.12)-(5.13)-(5.14)),  $j = i$ 
 $\forall R_j \in W(R_i)(\tau_k)$ 
    Demander  $f_j^C(t)$  et  $[x_j, y_j]^T$  (équations (5.12)-(5.13)-(5.14))
    Calculer  $\mathcal{F}_{ij}$  (équations (5.12)-(5.13)-(5.14))
    si  $\mathcal{F}_{ij} > \alpha \mathcal{F}_{ii}$ , ajouter  $R_j$  dans  $\mathcal{C}(R_i)$ ,  $\alpha \in [0, +\infty[$ 
fin
si  $\text{card}\mathcal{C}(R_i) \neq 0$ 
     $\begin{cases} x_{G_i} = x_{G_p} \\ y_{G_i} = y_{G_p} \end{cases}$ 
    avec  $R_p$  le robot avec le plus grand  $\mathcal{F}_{ij}$ 
sinon
     $R_i$  ne coopère pas

```

TABLE 5.3 – *Processus de décision de coopération pour le robot R_i*

5.2.3 Modification des contraintes

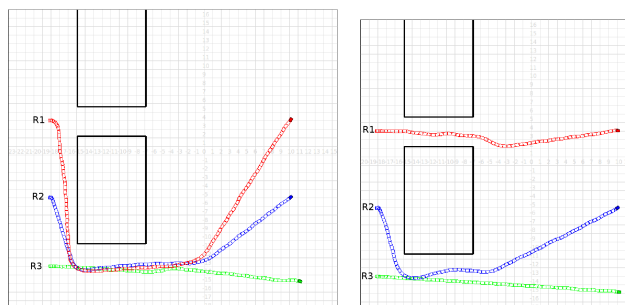
Il est aussi possible de changer les contraintes au cours de la tâche de navigation. Par exemple :

- Il est possible d'ajouter des zones avec des limitations de vitesse où le robot doit diminuer ou augmenter sa vitesse,
- Il est possible d'ajouter des zones interdites au robot,
- Il est possible d'ajouter des zones de l'espace où la distance minimum entre le robot et l'obstacle ou entre deux robots est diminuée ou augmentée.

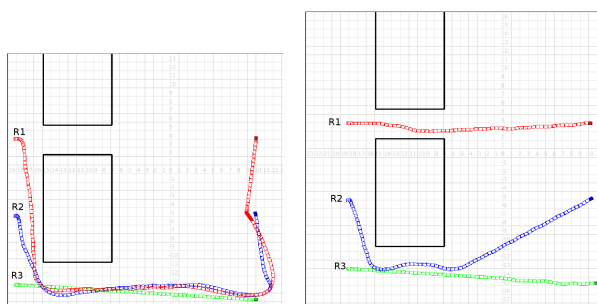
Remarque 5.4 *De manière générale, il est possible d'ajouter n'importe quelle contrainte pouvant s'exprimer comme une fonction différentielle des sorties plates.*



(a) Sans coopération, $\alpha \gg 1$



(b) Avec coopération $\beta = 0$, $\alpha = 1$ (c) Avec coopération $\beta = 0$, $\alpha = 1.5$



(d) Avec coopération $\beta = 1$, $\alpha = 1$ (e) Avec coopération $\beta = 1$, $\alpha = 1.5$

FIGURE 5.6 – *Coopération avec changement d'objectif - effet sur la trajectoire*

La figure 5.7 montre un exemple où il y a trois zones avec des limitations de vitesse différentes. Il est bien visible que la vitesse du robot diminue ou augmente (les points sont pris à intervalles de temps réguliers, la distance entre deux points consécutifs donne donc une image de la vitesse du robot).

5.2.4 Modification simultanée des contraintes et de la fonction coût

Un dernier exemple, reprenant celui-ci la modification simultanée, de la fonction coût et des contraintes, est donné dans cette sous-section. Les limitations de vitesse varient en fonction des zones et des obstacles. La figure 5.8(a) illustre les résultats obtenus en utilisant comme fonction coût L_2 . Le comportement du robot est dangereux à cause de la proximité avec les obstacles. La figure 5.8(b) montre le comportement du robot quand les poids μ_i , de la fonction coût $\bar{L}_{(R_i)}$, (5.11), sont fixés à $\mu_1(t) = \mu_3(t) = \mu_5(t) = 0$, $\mu_2(t) = 1 - \mu_4(t)$, et le paramètre de sécurité de non-collision entre le robot et les obstacles est $\mu_4(t) = 0$ dans la zone de basse vitesse et $\mu_4(t) = 0.5$ dans la zone de haute vitesse. Dans ce cas, le robot est autorisé à avoir un comportement plus “dangereux” quand sa vitesse est faible.

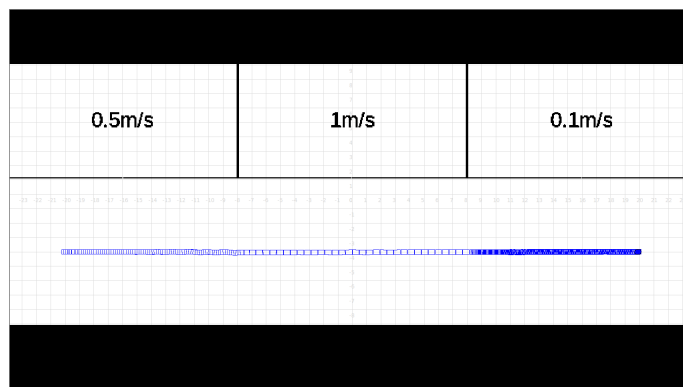


FIGURE 5.7 – Modifications des limitations de vitesse - effet sur la trajectoire

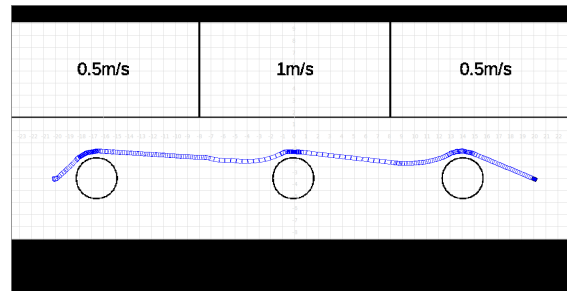
La figure 5.8(c) présente le comportement du robot quand le poids $\mu_4(t)$ a la même valeur durant toute la mission du robot, $\mu_4(t) = 0.5$ et $\mu_1(t) = \mu_3(t) = \mu_5(t) = 0$, $\mu_2(t) = 1 - \mu_4(t)$. Dans ce cas, le comportement du robot est sûr sur l'ensemble de la mission. Cette expérimentation montre que le comportement du robot change en fonction de la valeur des poids $\mu_4(t)$. Pendant la mission du robot, il est possible de définir un comportement sûr et un comportement dangereux en fonction de l'environnement.

5.2.5 Sur l'influence des différents paramètres

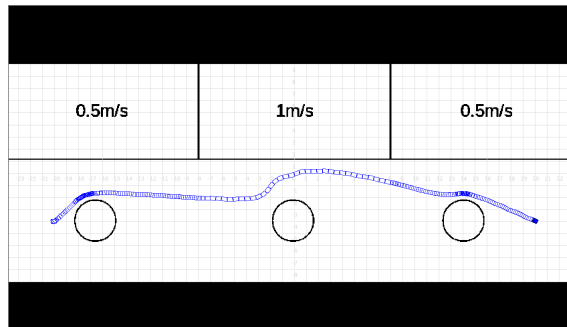
Dans cette section l'influence de plusieurs paramètres, sur le comportement d'un robot de la flottille a été étudiée. La table 5.4 reprend ces différents paramètres et en rappelle la signification.

Paramètre	Valeur	Signification
β	$\in [0, +\infty]$	COOPÉRATION : donne l'importance de la fonction de congestion de l'espace par rapport à la fonction d'accomplissement de la tâche (valeur typique $\in [0 10]$)
α	$\in [0, +\infty]$	COOPÉRATION : donne le comportement de coopération (voir la section 5.2.2.2) (valeur typique $\in [0 2]$)
μ_1	$\in [0, 1]$	FONCTION COÛT : importance de la \mathcal{L}_1 norme de la distance à l'objectif
μ_2	$\in [0, 1]$	FONCTION COÛT : importance de la \mathcal{L}_2 norme de la distance à l'objectif
μ_3	$\in [0, 1]$	FONCTION COÛT : état de charge de la batterie
μ_4	$\in [0, 1]$	FONCTION COÛT : paramètre de sécurité de non-collision avec les obstacles
μ_5	$\in [0, 1]$	FONCTION COÛT : paramètre de sécurité de non-collision avec les autres robots
		$\sum_{k=1}^5 \mu_k(t) = 1$

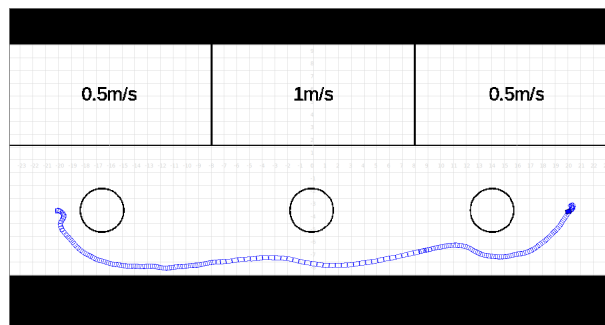
TABLE 5.4 – Choix des paramètres



(a) Comportement dangereux



(b) Comportement de sûreté différent en fonction des limitations de vitesses



(c) Comportement sûr dans toutes les zones de l'espace

FIGURE 5.8 – *Changement simultané de la fonction coût et des contraintes - Effet sur la trajectoire*

5.3 RÉSULTATS D'EXPÉRIMENTATIONS ET DE SIMULATIONS

Dans ce chapitre, un algorithme de stratégie de navigation a été proposé et son articulation avec le bloc de planification a été étudiée. La planification de trajectoire se base sur la minimisation d'une fonction coût sous un ensemble de contraintes provenant soit de la physique (limitation sur les vitesses par exemple) soit de notre problématique de coopération. Profitant des propriétés de platitude du modèle du robot en utilisant des fonctions coût et des contraintes s'exprimant comme des fonctions différentielles des sorties plates, le problème initial se ramène à un problème de calcul variationnel dont z est la variable de décision. Une paramétrisation de cette variable permet alors finalement de ramener ce dernier problème à un simple problème de programmation non linéaire. Le bloc de stratégie est présenté en détail dans la section précédente. Il est ainsi possible de faire varier au cours du temps la fonction coût, l'objectif intermédiaire en coopérant avec les voisins, ainsi que les contraintes et paramètres des contraintes. Dans cette

section, les algorithmes seront testés et leur efficacité sera étudiée à travers quatre critères : l'extensibilité, la complexité, le choix des paramètres et l'adaptabilité.

Cette section commencera donc par une présentation de ces critères. Ensuite, nous donnerons des résultats de simulations et des expérimentations afin d'illustrer la pertinence des algorithmes par rapport à ces critères.

5.3.1 Définition des critères de validation

5.3.1.1 L'extensibilité

Un algorithme de navigation est dit extensible si et seulement si il est possible d'augmenter le nombre de robots sans perte d'efficacité.

De notre point de vue, et dans l'optique d'être le plus proche possible de la réalité d'une flottille de robots sans meneur (nous avons expliqué en introduction la complexité inhérente à une approche sans meneur), être capable de faire naviguer une flottille de 10 robots ou plus sans meneur est un critère d'extensibilité raisonnable (à l'heure actuelle c'est une limite difficilement franchissable pour la plupart des algorithmes existants).

Nous garderons donc la formulation suivante pour le critère d'extensibilité : Un algorithme de planification de trajectoire pour une flottille de robots autonomes sans meneur est extensible si et seulement si :

- Chaque robot de la flottille termine sa propre mission, même si la flottille est composée d'un nombre de robots supérieur ou égal à 10,
- L'évolution du temps de navigation (*i.e.* le temps nécessaire pour que chaque robot de la flottille termine sa mission) doit être le plus proche possible d'une fonction linéaire du nombre de robots composant la flottille.

5.3.1.2 La complexité

La notion de complexité permet de quantifier le temps de calcul nécessaire pour obtenir une trajectoire optimale pour un robot. L'objectif est de montrer comment la complexité augmente en fonction du nombre de robots. La complexité doit être la plus proche possible d'une fonction linéaire du nombre de robots.

5.3.1.3 Choix des paramètres

Dans l'algorithme de coopération proposé, il n'y a que deux paramètres à régler, α , le paramètre de coopération et β , le poids de la fonction d'accomplissement de la tâche par rapport à la fonction de congestion de l'espace. Les résultats en simulation montrent comment le choix de ces paramètres influence le temps de navigation nécessaire pour que chaque robot termine sa mission.

5.3.1.4 L'adaptabilité de l'algorithme

Le critère d'adaptabilité de l'algorithme représente la capacité de l'algorithme à s'adapter à de nouveaux *scenarii* en nécessitant le moins de changements possibles. Moins il y a de changements, plus l'algorithme peut être considéré comme adaptable.

5.3.2 Résultats d'expérimentations et de simulations

5.3.2.1 Description de la plateforme

Le groupement de logiciels et bibliothèques de développement Player/Stage (voir [26]) a été utilisé tant pour les simulations que pour les expérimentations. Cet ensemble de logiciels permet d'avoir une bonne représentation (dans le cadre des simulations) de la réalité physique d'une flottille de robots, notamment en matière de temps de communication (communication TCP/IP). Chaque robot a son propre processus qui communique avec les processus de ses voisins *via* le serveur de communication Player. Même si physiquement les différents processus sont exécutés sur le même ordinateur, il est possible de considérer que la flottille est décentralisée, car les processus sont réellement indépendants les uns des autres et n'ont pour lien que le serveur de communication. Il est possible d'envisager, dans le cadre de robots indépendants ayant chacun sa propre capacité de communication, d'utiliser des protocoles proches de ceux utilisés pour le partage de fichiers sur internet qui permettent de s'affranchir de la nécessité d'avoir un serveur central (voir par exemple les travaux de [114, 115]).

Dans cette partie, nous ne présentons que les travaux relatifs aux blocs stratégie (voir la section 5.2) et ceux relatifs au bloc de planification de trajectoire (voir la section 5.1). Les paramètres nécessaires pour l'algorithme de planification de trajectoire sont rappelés dans la table 5.5

Paramètres	Expérimentation	Simulation
v_{max}	$0.1m/s$	$0.8m/s$
ω_{max}	$4 \times rad/s$	$4 \times rad/s$
a_{max}	$0.5m/s^2$	$2m/s^2$
vision range	$1m$	$5m$
taille du voisinage	$2m$	$10m$
taille de l'environnement	$2m \times 3m$	$400m^2$
$d(R_i, R_j)$	$\geq 0.1m$	$\geq 0.1m$
Temps d'échantillonnage T_s	$0.15s$	$0.15s$
T_p	$80 \times T_s$	$80 \times T_s$
Période de planification T_c	$2 \times T_s$	$2 \times T_s$
$\mu_2(t)$	1	1
$\mu_1(t) = \mu_3(t) = \mu_4(t) = \mu_5(t)$	0	0
α	$\gg 1$	variant
β	non utilisée	variant

TABLE 5.5 – Paramètres utilisée pour les simulations et les expérimentations

Les données issues des blocs de perception et de localisation sont simulées en utilisant «Stage» (pour les simulations) et sont calculées en utilisant un système de localisation composé d'une caméra au dessus du plan de jeu (pour l'expérimentation). Dans les deux cas, ces données sont envoyées au robot *via* «Player». Le bloc de suivi de trajectoire n'a pas été implémenté, la trajectoire étant très régulièrement recalculée (paramètre T_c) à partir de la position réelle du robot. Le bloc de retour actionneur n'est pas nécessaire en simulation et est implémenté directement dans le robot pour les expérimentations. Le robot mobile unicycle utilisé pour les simulations est un robot Pioneer de la société Adept mobilerobots. Il communique avec Player *via* une communication TCP/IP.

Le robot utilisé pour les expérimentations est un Miabot de la société Merlin Robotics. Ce robot de $8cm \times 8cm$ est aussi un robot unicycle et la communication se fait *via* le protocole *Bluetooth*. La communication avec Player est assurée par un driver.

La figure 5.9 présente la carte pour les simulations et l'environnement pour les expérimentations. La taille de l'environnement pour la simulation est de $400m^2$ avec de nombreux obstacles de grande taille par rapport à la vision du robot (limitée à $5m$). Chaque robot de la flottille connaît sa position initiale et sa position finale. La position finale du robot $R_i, i \in \{1, 2, \dots, 8\}$ est la position symétrique de son point de départ par rapport à l'axe vertical passant par l'origine tandis que la position finale du robot $R_i, i \in \{9, 10, \dots, 16\}$ est la position symétrique de son point initial par rapport à l'axe horizontal passant par l'origine. Rappelons que l'origine de l'environnement est le point du milieu de ce dernier (cf. figure 5.9).

La taille de l'environnement pour les expérimentations est $2m \times 3m$ et il y a plusieurs obstacles de formes différentes. Cet environnement représente un espace urbain à échelle réduite. La position initiale est en face de la station de police pour les robots policiers ou en face de la banque pour le robot voleur. La position finale des robots dépend de la trajectoire du robot voleur (voir les règles du jeu dans la suite de la section).

5.3.2.2 Résultats pour le critère d'extensibilité

Les résultats de simulation ont démontré que l'algorithme n'est pas extensible si $\alpha \gg 1$ (il n'y a pas de coopération entre le robot et son voisinage) et que la flottille comporte plus de 7 robots. Dans ce cas, la flottille n'a pas réussi sa mission. Pour $\alpha = 1.5$, la flottille a réussi sa mission : il en sera de même pour des flottilles constituées jusqu'à 16 robots. Étant donné que la flottille travaille sans meneur, au delà de 17 robots le volume de données à traiter par le serveur de communication devient trop important pour réussir à mener à bien la mission de chaque robot. Une approche de communication différente devrait permettre de lever en partie ces problèmes.

Sur la figure 5.10 deux courbes sont représentées. La première en bleu donne le temps de navigation en fonction du nombre de robots quand les robots ne coopèrent pas entre eux. La seconde courbe en rouge donne la même information mais cette fois ci quand les robots coopèrent entre eux. Quand la flottille n'a qu'un petit nombre de robots (moins de 4), l'algorithme sans coopération est plus "efficace" que celui avec coopération, car dans ce cas, le volume de communication entre les robots est plus faible. Cependant, quand le nombre de robots augmente, la flottille n'arrive plus à remplir sa mission et l'algorithme avec coopération devient plus efficace. En effet, la complexité de la mission augmentant, les robots tirent bénéfice des nombreuses informations en provenance de leurs voisins. Concernant la forme de la courbe, la figure montre que le temps de navigation est proche d'une fonction linéaire par rapport au nombre de robots.

Remarque 5.5 Les valeurs des paramètres β et α sont $\beta = 5$ et $\alpha = 1$.

5.3.2.3 Résultats pour le critère de complexité

Les résultats présentés sur la figure 5.11 montrent l'évolution du temps moyen normalisé d'optimisation d'un morceau de trajectoire (sur un horizon de planification

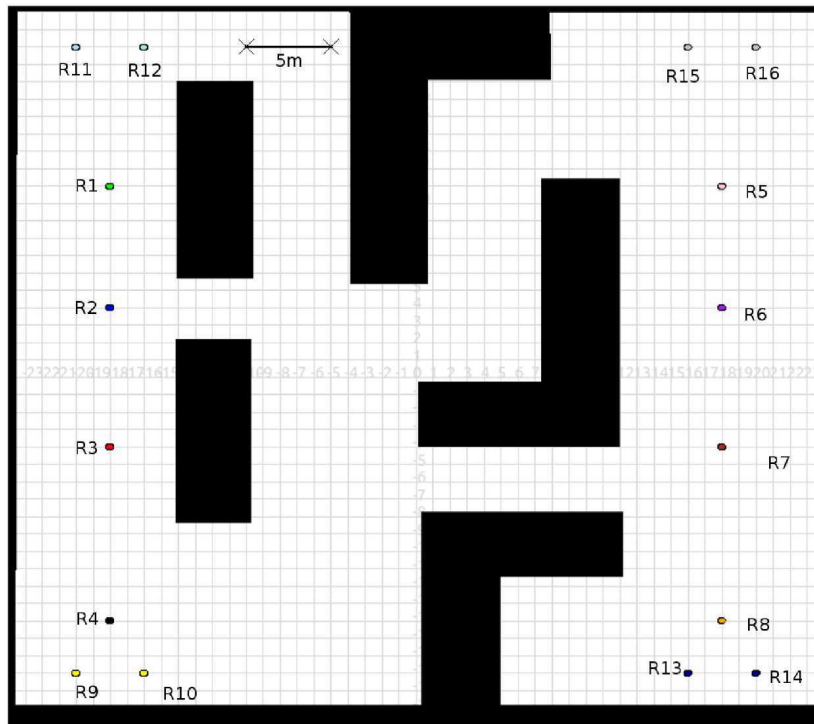


FIGURE 5.9 – Carte des environnements utilisés pour les simulations et pour les expérimentations

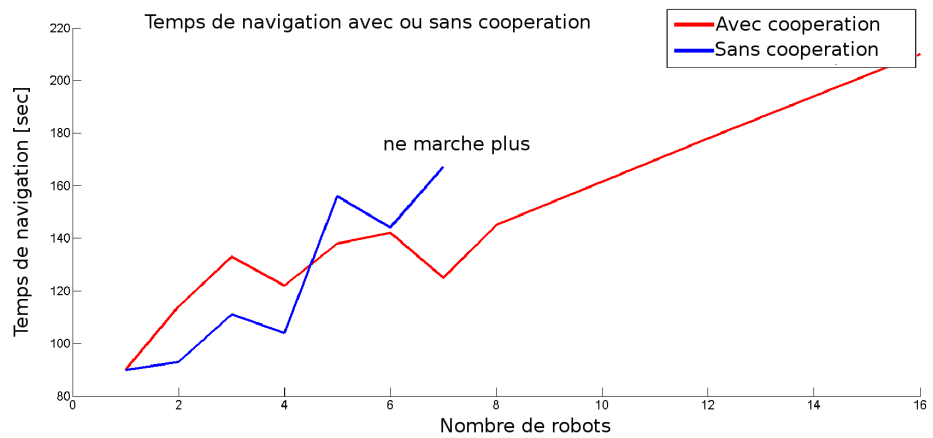


FIGURE 5.10 – Temps de navigation en fonction du nombre de robots dans la flottille, $\beta = 5, \alpha = 1$

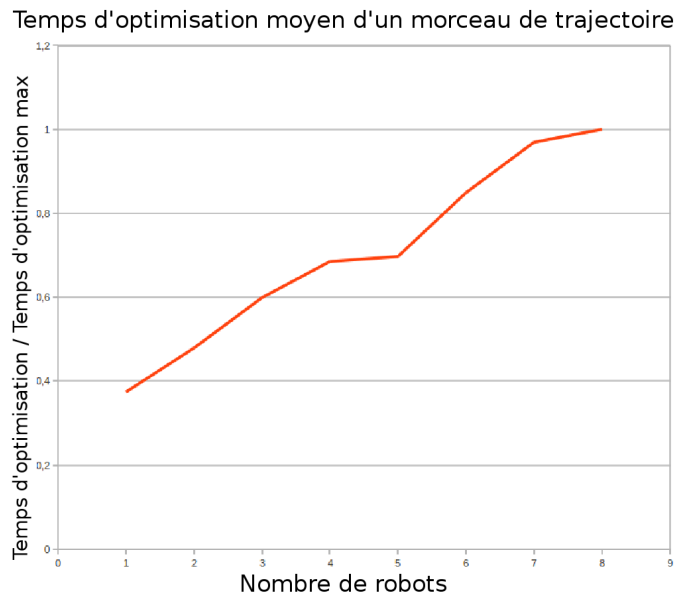


FIGURE 5.11 – Temps d'optimisation moyen en fonction du nombre de robots

	$\beta = 7$	$\beta = 5$
$\alpha = 1$	137,4s	152,8s
$\alpha = 1.2$	149s	167,6s

TABLE 5.6 – Temps de navigation en fonction de α , β

tion) pour un robot en fonction du nombre de robots (en effet, il faut se souvenir que chaque morceau de trajectoire est issu de l'optimisation d'une fonction coût). Ce temps moyen est la moyenne des temps d'optimisation sur le calcul d'un grand nombre de trajectoires divisé par le résultat obtenu pour le nombre maximum de robots. Ce temps moyen est proche d'une fonction linéaire du nombre de robots, ce qui valide notre critère de complexité.

5.3.2.4 Résultats concernant le choix des paramètres

Les résultats de simulation sont donnés sur la table 5.6. Comme attendu, la flottille termine sa mission plus rapidement quand la fonction d'accomplissement de la tâche est plus importante que la fonction de congestion de l'espace, *i.e.* quand β est plus grand que 1. De plus la flottille termine aussi plus rapidement sa mission quand les robots coopèrent plus les uns avec les autres *i.e.* quand α est plus petit (en restant supérieur à 1).

5.3.2.5 Critère d'adaptabilité

Pour prouver l'adaptabilité de l'algorithme à de nouveaux *scenarii*, l'algorithme proposé a été testé sur le scénario du jeu des policiers contre le voleur. Le robot voleur est conduit par un humain *via* un joystick (volant de jeu), il vient juste de «braquer» une banque (sa position initiale est donc face à celle-ci) et doit s'enfuir pour rejoindre l'aéroport. Les policiers (ici les quatre robots autonomes) quittent au début du jeu la station de police et doivent l'intercepter. Le voleur gagne le jeu s'il arrive à l'aéroport sans être attrapé par les policiers. Les policiers gagnent

Victoire des policiers	Victoire des voleurs	Temps moyen de l'expérience
43	7	2min 15sec

TABLE 5.7 – Résultats expérimentaux

le jeu s'ils arrivent à entourer le voleur avant qu'il n'arrive à l'aéroport. Tous les policiers sont autonomes, ainsi la planification de trajectoire est faite en ligne (chaque robot possédant son propre processus). Chaque policier communique avec ses voisins pour planifier une trajectoire sans collisions entre robots. Ils ne connaissent que leurs positions, la position du voleur, et la position des obstacles qui sont dans l'ensemble $\mathcal{O}_i(t)$. Ces informations sont données par un système de localisation composé d'une caméra au dessus du plan de jeu. A chaque période d'échantillonnage T_c , les policiers calculent une trajectoire afin d'entourer le voleur. La trajectoire est régulièrement mise à jour, pour prendre en compte les changements de trajectoire du voleur.

La seule différence dans l'architecture logicielle entre les simulations et les expérimentations réside dans le choix de l'objectif final. Alors que pour les simulations l'objectif final était fixé au début de l'expérience, ici à chaque période d'échantillonnage les policiers doivent changer leur objectif pour venir entourer le voleur. Pour tout le reste l'architecture logicielle reste identique. L'expérience a été effectuée 50 fois, les résultats sont résumés dans la table 5.7. La capacité de l'architecture logicielle de s'adapter à une situation hautement dynamique, où l'objectif bouge sans cesse en temps réel, montre l'aptitude de celle-ci à répondre aux besoins d'une flottille décentralisée en terme de navigation autonome sans collision et respectant les contraintes physiques des robots et les besoins de temps réel pour l'application.

Une vidéo est disponible à la page web suivante

<http://researchers.lille.inria.fr/~jrichard/video/robotcity.wmv>

CONCLUSION DU CHAPITRE

Ce chapitre propose une un organigramme fonctionnel d'un robot appartenant à une flottille décentralisée de robots mobiles qui doivent naviguer de façon autonome dans un environnement partiellement connu avec des obstacles. Nous avons ici particulièrement étudié le bloc stratégie qui est le cœur de ce chapitre. Durant la tâche de navigation, le bloc stratégie d'un robot récupère les informations en provenance du bloc localisation et perception des autres robots (et éventuellement de l'opérateur) *via* le bloc de communication. Il traite alors l'ensemble de ces données pour transmettre au bloc de planification de trajectoire les informations nécessaires pour pouvoir planifier une trajectoire sans collision et respectant le modèle cinématique du robot et ses contraintes physiques.

Grâce à cette architecture il est possible de changer en temps réel, durant la tâche de navigation, la fonction coût, les contraintes et l'objectif du robot.

Cinq fonctions coût différentes sont proposées et un algorithme de fusion utilisant une somme pondérée de celles-ci est donné. L'influence des différents poids sur la forme de la trajectoire du robot est étudiée à travers des simulations. Dans l'optique de finir sa mission, chaque robot peut coopérer avec ses voisins pour obtenir des informations complémentaires sur son environnement, chaque robot

transmettant alors aux autres ses propres informations locales issues de son bloc perception et localisation. Le processus de coopération est explicité et testé en simulation pour mettre en évidence l'influence des différents paramètres.

Finalement quatre critères permettant d'étudier à travers des simulations et des expérimentations les caractéristiques de l'algorithme sont présentés. Ainsi, l'extensibilité, la complexité, le choix des paramètres et l'adaptabilité sont étudiés.

Les trois premiers paramètres sont validés en simulation avec une flottille de robots allant jusqu'à 16 robots. Une expérimentation est effectuée en utilisant le scénario du jeu du voleur et des policiers pour valider l'adaptabilité de l'approche proposée.

CONCLUSION GÉNÉRALE

6

6.1 CONCLUSION

Nous avons dans ce manuscrit étudié deux problématiques majeures de la robotique mobile, la localisation et la planification de trajectoire, et ce à travers le prisme de l'algèbre différentielle. L'algèbre différentielle dont les principaux concepts nécessaires à notre étude sont rappelés dans le chapitre 2 est particulièrement adaptée à l'étude des systèmes non-linéaires. L'un des avantages fondamentaux des robots mobiles à roues est qu'ils appartiennent tous, comme cela a été rappelé dans le chapitre 2, à l'une des cinq classes de robots mobiles. Il est donc envisageable d'effectuer une étude exhaustive de ces deux problématiques pour l'ensemble des robots mobiles à roues. Pour conclure le chapitre 2, nous avons présenté un différenciateur numérique en environnement bruité qui est à la base des estimateurs de posture proposés dans les deux chapitres suivants.

Les chapitres 3 et 4 s'attaquent à la problématique de la localisation, de robot unicycle pour le premier et de n'importe quel type de robot pour le deuxième. Notre étude se place dans le cadre de la localisation par mesure angulaire seule, un capteur typique de ce cas d'étude est par exemple une caméra monoculaire. Aucune hypothèse sur la nature des bruits pouvant perturber les mesures n'est faite. Nous présentons d'abord le concept de localisabilité, lui-même extrêmement lié à la notion d'observabilité. L'observabilité des systèmes non-linéaires et donc la localisabilité des robots mobiles peut être étudiée dans le cadre de l'algèbre différentielle puisqu'il suffit de trouver une relation algèbro-différentielle entre les sorties mesurées et la posture pour prouver la localisabilité. L'avantage de l'approche algébrique par rapport à une approche géométrique est sa capacité à donner en plus du critère de localisabilité un estimateur de posture basé sur la relation algèbro-différentielle. Nous effectuons l'étude pour chacun des robots dans cinq cas d'étude différents. Dans chacun de ces cas, le robot observe les angles relatifs entre lui-même et un ou deux points d'intérêts. Dans le chapitre 3 nous montrons qu'il est possible de fusionner les différentes estimations lorsque le nombre de points d'intérêt disponibles est supérieur à celui nécessaire pour la localisabilité. Dans ces deux chapitres des résultats de simulations sont donnés et, dans le cadre du robot unicycle, les résultats obtenus sont comparés avec les résultats donnés par le filtre de Kalman Étendu pour les mêmes situations. Nous montrons que les résultats obtenus se comparent favorablement avec cet estimateur, et ce avec des hypothèses d'entrées plus faibles. Nous effectuons ensuite des expérimentations dans l'une des hypothèses de mesure qui permettent de valider sur un cas réel les algorithmes proposés. La solution proposée dans ce travail présente plusieurs points d'intérêt et d'amélioration par rapport à la littérature :

1. Nos algorithmes peuvent donner un critère de localisabilité, lorsque le robot n'est pas localisable il est possible de connaître le nombre d'informations manquantes pour pouvoir localiser le robot,
2. Contrairement à l'EKF, il n'est pas nécessaire d'initialiser l'algorithme,
3. Contrairement à l'EKF, il n'est pas nécessaire de connaître les propriétés statistiques du bruit,
4. Contrairement à l'EKF, il n'est pas nécessaire de connaître les commandes envoyées au robot, ces dernières peuvent même dans certains cas être estimées par l'estimateur proposé.

Dans le chapitre 5 nous présentons une articulation entre les blocs de stratégie et de planification de trajectoire qui permet de tirer parti de la dualité discret/continu. Le bloc stratégie transmet au bloc de planification de trajectoire une fonction coût, les paramètres des contraintes de planification, ainsi que l'objectif final ou intermédiaire. L'ensemble de ces paramètres peut être changé en temps réel durant la mission, permettant donc de prendre en compte l'évolution de l'état interne du robot, les informations reçues de ses voisins ainsi que les modifications de l'environnement perçues par les capteurs du robot. La fonction coût proposée dans ce travail est la somme pondérée par des coefficients variables de cinq fonctions coûts unitaires. Une étude des coefficients et de leur influence sur la trajectoire empruntée par le robot est faite à travers des simulations sur des cas simples et illustratifs. Le deuxième élément qui peut être transmis entre le bloc de stratégie et le bloc de planification est un objectif intermédiaire. Cette objectif est choisi en fonction de l'objectif final du robot, et des informations en provenance des voisins afin de maximiser l'espace libre autour du robot et de faciliter sa mission globale. Le troisième élément à être transmis, ce sont les contraintes et leurs paramètres, nous illustrons à travers un exemple simple cette possibilité. Finalement, nous effectuons une étude en simulation et sur un cas réel de l'ensemble de la solution proposée et étudions ses caractéristiques sur quatre critères d'études, l'extensibilité, la complexité, le choix des paramètres et l'adaptabilité de l'algorithme. L'expérimentation est menée sur une plateforme de démonstration mise en place par l'INRIA sur leur salle d'exposition à Euratechnologie. Cette démonstration à destination notamment du grand public nous a permis de tester nos algorithmes dans une situation temps réel hautement dynamique.

6.2 PERSPECTIVES

Le travail présenté dans ce manuscrit prend corps sur les techniques issues de l'algèbre différentielle et notamment sur un différentiateur numérique en environnement bruité. Ce différentiateur développé dans le cadre du projet NON-A à l'INRIA est amené à évoluer et toute amélioration de ses caractéristiques permettra l'amélioration des estimateurs présentés dans ce travail,

Nous avons présenté cinq classes de robots mobiles mais avons effectué l'étude de la localisabilité et de la localisation uniquement pour quatre de ces cinq classes. Il faudrait étendre les résultats obtenus pour la cinquième classe,

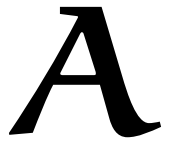
Notre étude a été menée sur le cas mesure angulaire seule (un capteur typique de cette situation étant une caméra monoculaire). Il serait intéressant d'étendre notre étude sur d'autres modèles de mesure, notamment de distance avec comme capteur typique un capteur laser,

Nous avons étudié le cas des robots mobiles à roues mais il serait possible d'utiliser la même approche pour d'autres types de robots mobiles comme les drones par exemple où le problème de la localisation est encore difficile à résoudre,

Nous avons principalement étudié le problème de la localisation, il serait intéressant d'étendre cette approche au problème du SLAM où la carte et la localisation du robot sont à construire simultanément,

Finalement, l'algorithme de planification et de stratégie proposé doit être étendu à une plus large classe de situation pour bien prouver son adaptabilité et ses performances.

ANNEXES



SOMMAIRE

A.1	RAPPEL D'ALGÈBRE	157
A.1.1	Définitions générales	157
A.2	ÉTUDE DE L'OBSERVABILITÉ DANS LE CADRE GÉOMÉTRIQUE	160
A.3	CROCHET DE LIE	162
A.4	INTERSECTION DE DEUX CERCLES	163
A.5	ÉTUDE DE LA PLATITUDE À L'AIDE DE MAPLE	164
A.5.1	Platitude d'un robot de type $(2, 0)$	164
A.5.2	Platitude d'un robot de type $(3, 0)$	165
A.5.3	Platitude d'un robot de type $(2, 1)$	165
A.5.4	Platitude d'un robot de type $(1, 1)$	165

A.1 RAPPEL D'ALGÈBRE

Nous allons dans cette section effectuer un certain nombre de rappels sur l'algèbre différentielle. Pour une revue plus exhaustive, le lecteur pourra se reporter au chapitre 6 de [129].

A.1.1 Définitions générales

Définition A.1 (Monoïde) *Un monoïde M est un ensemble non vide muni d'une opération $*$: $M \times M \rightarrow M$ qui :*

- est associative : $a * (b * c) = (a * b) * c$ pour tout $a, b, c \in M$.
- possède un élément neutre : il existe un élément unique $e \in M$ tel que :

$$e * a = a = a * e, \forall a \in M. \quad (\text{A.1})$$

- Exemple A.1**
- L'ensemble des entiers naturels muni de l'addition est un monoïde, d'élément neutre 0,
 - L'ensemble des entiers naturels muni de la multiplication est un monoïde d'élément neutre 1,
 - L'ensemble des entiers naturels muni de la loi \max qui à deux entiers associe le plus grand des deux est un monoïde d'élément neutre 0,

Définition A.2 (Un groupe) *Un groupe est un ensemble tel que $(G, *)$ soit un monoïde et tel que tout élément de G ait un élément inverse, c'est-à-dire que pour tout $g \in G$, il existe $h \in G$ tel que $g * h = e = h * g$. Il est dit **abélien** ou **commutatif** si la loi interne est commutative.*

- Exemple A.2** *L'ensemble des entiers relatifs \mathbb{Z} muni de la somme usuelle est un groupe d'élément neutre 0.*

Définition A.3 (Un anneau) *Un anneau R est un ensemble muni de deux opérations $+$: $R \times R \rightarrow R$ et \cdot : $R \times R$ tel que $(R, +)$ soit un groupe abélien, (R, \cdot) soit un monoïde et tel que la deuxième loi soit distributive par rapport à la première (c'est-à-dire que $a \cdot (b + c) = a \cdot b + a \cdot c$ et $(a + b) \cdot c = a \cdot c + b \cdot c$ pour tout $a, b, c \in R$).*

Définition A.4 (Anneau Commutatif) *Si $a \cdot b = b \cdot a$, alors R est appelé un **anneau commutatif***

- Exemple A.3**
- L'ensemble des entiers relatifs $(\mathbb{Z}, +, \cdot)$ est un anneau commutatif d'élément neutre 0 pour la somme (usuelle) d'élément neutre 1 pour la multiplication (usuelle),
 - L'ensemble des polynômes à coefficients complexes $(\mathbb{C}[X], +, \cdot)$ est un anneau commutatif,
 - L'ensemble des matrices 2×2 à valeurs réelles est un anneau non commutatif.

Définition A.5 (Diviseur de zéro) *On dit qu'un anneau R n'a pas de **diviseur de zéro** si $a \cdot b = 0$ implique $a = 0$ ou $b = 0$ pour tout $a, b \in R$*

- Exemple A.4**
- L'anneau des entiers relatifs $(\mathbb{Z}, +, \cdot)$ est un anneau n'ayant pas de diviseur de 0,
 - L'anneau des polynômes à coefficients complexes $(\mathbb{C}[X], +, \cdot)$ est un anneau n'ayant pas de diviseur de 0,

- L'anneau des matrices 2×2 , $(\mathcal{M}_{2 \times 2}, +, \cdot)$ possède des diviseurs de 0. En effet :

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \quad (\text{A.2})$$

Définition A.6 (Domaine intégral) *Un anneau commutatif R sans diviseur de zéro est appelé un **domaine intégral**.*

- Exemple A.5**
- L'anneau des entiers relatifs $(\mathbb{Z}, +, \cdot)$ est un domaine intégral,
 - L'anneau des polynômes à coefficients complexes $(\mathbb{C}[X], +, \cdot)$ est aussi un domaine intégral.

Définition A.7 (Homomorphisme d'anneau) *Un **homomorphisme d'anneau** $\varphi : R \rightarrow R$ est une application qui préserve les opérations, c'est-à-dire telle que :*

- $\varphi(a + b) = \varphi(a) + \varphi(b)$
- $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$

Définition A.8 (Idéal) *Un sous-ensemble I d'un anneau commutatif R est un **idéal** de R si :*

- $a - b \in I$ pour tout $a, b \in I$
- $ra \in I$ pour tout $r \in R, a \in I$

Exemple A.6 *L'ensemble $I = \{X \times P(X) \mid P(X) \in \mathbb{C}[X]\}$ est un idéal de $\mathbb{C}[X]$.*

Définition A.9 (Anneau quotient) *Prenons I un idéal d'un anneau commutatif R . Définissons une relation d'équivalence sur R par :*

$$a \sim b \Leftrightarrow a - b \in I \quad (\text{A.3})$$

Notons par $a + I$ la classe d'équivalence de $a \in R$. **L'anneau quotient** R/I est défini par :

$$R/I = \{a + I \mid a \in R\} \quad (\text{A.4})$$

Exemple A.7 *Prenons I l'idéal défini dans l'exemple A.6. C'est un idéal de l'anneau commutatif $\mathbb{C}[X]$. Alors $\mathbb{C}[X]/I \sim \mathbb{C}$. En effet prenons un polynôme de $\mathbb{C}[X]$:*

$$P(X) = a_0 + a_1X + \dots + a_nX^n$$

Alors

$$\begin{aligned} P(X) &= a_0 + X(a_1 + \dots + a_nX^{n-1}) \\ &= a_0 + Q(X), Q(X) \in I \end{aligned}$$

Définition A.10 (Idéal premier) *Prenons I un idéal d'un anneau commutatif $R, I \neq R$. Nous appelons I un **idéal premier** si pour tout $a, b \in R$:*

$$a \cdot b \in I \Rightarrow a \in I \text{ ou } b \in I \quad (\text{A.5})$$

Exemple A.8 *L'idéal I défini dans l'exemple A.6 est un idéal premier. En effet, prenons deux éléments $\{P(X), Q(X)\}$ de $\mathbb{C}[X]$ tels que $P(X) \times Q(X) \in I$. Prenons :*

$$\begin{aligned} P(X) &= p_0 + p_1X + \dots + p_nX^n \\ Q(X) &= q_0 + q_1X + \dots + q_mX^m \end{aligned}$$

Alors

$$P(X)Q(X) = p_0q_0 + (p_1q_0 + p_0q_1)X + \dots$$

Et donc si $P(X)Q(X) \in I$ alors $p_0q_0 = 0$ et donc soit $p_0 = 0$ soit $q_0 = 0$ soit $p_0 = q_0 = 0$ et donc l'un des deux polynômes au moins appartient à I .

Proposition A.1 Prenons R un anneau commutatif et I un idéal de R , $I \neq R$. Alors :

I est **premier** si et seulement si R/I est un **domaine intégral**.

Exemple A.9 D'après l'exemple A.8, en utilisant l'idéal I défini dans l'exemple A.6, $\mathbb{C}[X]/I$ est un domaine intégral.

Définition A.11 (Inversible) Prenons R un anneau commutatif. Un élément $a \in R$ est **inversible** s'il existe $b \in R$ tel que $a \cdot b = 1$.

Définition A.12 (Corps) Un **corps** K est un anneau commutatif pour lequel tout élément non nul est inversible

Exemple A.10 L'ensemble des fractions de polynômes à coefficients complexes (noté $\mathbb{C}(X)$) forme un corps. En effet, $\forall P(X) \in \mathbb{C}[X]$, $P(X) \times \frac{1}{P(X)} = 1$ avec $\frac{1}{P(X)} \in \mathbb{C}(X)$

Définition A.13 (Le corps des fractions) Prenons R un anneau commutatif sans diviseur de zéro. Définissons une relation d'équivalence sur $R \times R$ par

$$(a, b) \sim (c, d) \Rightarrow a \cdot d = b \cdot c \quad (\text{A.6})$$

Notons $\frac{a}{b}$ la classe d'équivalence de (a, b) . Le corps des fractions de R est l'ensemble :

$$\text{Fract}(R) = \left\{ \frac{a}{b} \mid a, b \in R, b \neq 0 \right\} \quad (\text{A.7})$$

Définition A.14 (Espace vectoriel) Un **espace vectoriel** V sur un corps K est un ensemble muni de deux opérations

$$+ : V \times V \rightarrow V$$

Et

$$\begin{aligned} K \times V &\rightarrow V \\ (\lambda, v) &\mapsto \lambda v \end{aligned}$$

Tel que :

- $(V, +)$ est un groupe abélien
- distributivité : $\lambda \cdot (u + v) = \lambda \cdot u + \lambda \cdot v$ et $(\lambda + \mu) \cdot v = \lambda v + \mu v$ pour tout $\lambda, \mu \in K$ et $u, v \in V$
- associativité : $\lambda \cdot (\mu \cdot v) = (\lambda\mu) \cdot v$ pour tout $\lambda, \mu \in K, v \in V$
- $1 \cdot v = v$ pour tout $v \in V$

Exemple A.11 L'ensemble $\mathbb{C}[X]$ muni des opérations usuelles est un \mathbb{C} -espace vectoriel.

Définition A.15 (Linéairement (in)dépendante / Famille génératrice / Base) Prenons V un K -espace vectoriel, I un ensemble d'indice, et $\mathcal{B} = \{v_i\}_{i \in I}$ une famille d'éléments de V .

– \mathcal{B} est **linéairement indépendante** si toute combinaison de la forme

$$\lambda_1 v_{i_1} + \dots + \lambda_n v_{i_n} = 0 \quad (\text{A.8})$$

Implique

$$\lambda_1 = \lambda_2 = \dots = \lambda_n = 0 \quad (\text{A.9})$$

– \mathcal{B} est **linéairement dépendante** si elle n'est pas linéairement indépendante
 – \mathcal{B} est une **famille génératrice** de V si tout élément de V peut être réécrit comme une combinaison linéaire de la forme :

$$\lambda_1 v_{i_1} + \dots + \lambda_n v_{i_n}, (\lambda_i \in K, v_{i_k} \in V) \quad (\text{A.10})$$

– Une famille, sous-ensemble de V , linéairement indépendante et qui génère V est appelée une **base** de V

Exemple A.12 Prenons le \mathbb{C} –espace vectoriel $\mathbb{C}[X]$:

- La famille $\{1, X, X^2\}$ est une famille linéairement indépendante (ou libre) de $\mathbb{C}[X]$,
- La famille $\{1, X, 2X\}$ est une famille linéairement dépendante de $\mathbb{C}[X]$,
- La famille $\{1, X, \dots, X^n, \dots\}$ est une base de $\mathbb{C}[X]$ de cardinal infini.

A.2 ÉTUDE DE L'OBSERVABILITÉ DANS LE CADRE GÉOMÉTRIQUE

Soit le processus physique représenté sous la forme suivante :

$$\begin{cases} \dot{x} = f(x(t), u(t)) \\ y = h(x(t)) \end{cases} \quad (\text{A.11})$$

L'observateur est un dispositif permettant d'estimer l'état initial x à partir de la sortie mesurée y et de la commande u . Si le système est complètement observable, il est alors possible d'estimer l'état initial. S'il ne l'est pas, alors il est impossible d'avoir une information complète de l'état. La notion d'observabilité est liée à la notion d'indiscernabilité définie de la façon suivante :

Définition A.16 (Indiscernabilité, [10]) *Un état $x(0)$ est **indiscernable** d'un autre état $x'(0)$ pour le système (A.11) si*

$$\forall u \in \mathcal{U}, \forall t \geq 0, y(t, 0, x(0), u) = y(t, 0, x'(0), u), \quad (\text{A.12})$$

où $y(t, 0, x(0), u) = h(x(t, 0, x(0), u))$ (resp. $y(t, 0, x'(0), u) = h(x(t, 0, x'(0), u))$) est la sortie du système (A.11) pour l'entrée u et pour l'état initial $x(0)$ (resp. $x'(0)$).

Cette notion d'indiscernabilité de deux états initiaux permet de donner la définition de l'**observabilité** suivante :

Définition A.17 (Observabilité, [10]) *Un système (A.11) est dit **observable** en x s'il n'y a pas d'état indiscernable pour x dans \mathcal{X} . Le système (A.11) est observable si et seulement s'il est observable pour tout $x \in \mathcal{X}$.*

Cette définition est cependant trop générale. Parfois, il peut être suffisant d'arriver à distinguer un état de son voisinage. Prenons par exemple le système suivant :

$$\begin{cases} \dot{x} = u \\ y = \cos(x). \end{cases}$$

Clairement, y ne permet pas de faire la distinction entre par exemple $x_0 = 0$ et $x_0 = 2k\pi$, et donc le système est non observable. Cependant, y permet tout de même de faire la distinction entre les états sur l'intervalle $]-\frac{\pi}{2}, \frac{\pi}{2}[$.

Une notion plus faible de l'observabilité peut donc être définie de la façon suivante :

Définition A.18 (Observabilité locale, [10]) *Le système (A.11) est localement observable en x s'il existe un voisinage $\mathcal{V}(x)$ de x tel qu'il n'y a pas d'état indiscernable de x dans $\mathcal{V}(x)$. Le système (A.11) est localement observable s'il est localement observable pour tout $x \in \mathcal{X}$.*

Notons que deux états peuvent être indiscernables pour certaines entrées, alors qu'ils sont discernables pour d'autres. Considérons par exemple le système suivant :

$$\begin{cases} \dot{x}_1 = x_2 u \\ \dot{x}_2 = 0 \\ y = x_1. \end{cases}$$

Il est clair que pour l'entrée nulle u , il n'est pas possible de discerner les deux états x et x' tel que $x_1 = x'_1$ et $x_2 \neq x'_2$. Cependant le système est observable pour n'importe quel $u \neq 0$. Par exemple si $u = 1$, nous obtenons alors un système linéaire observable. C'est pourquoi afin d'éviter cette situation il est nécessaire d'affiner encore un peu la définition :

Définition A.19 (Observabilité faible, [10]) *Le système (A.11) est **faiblement observable** en x s'il existe un voisinage $\mathcal{V}(x)$ de x tel que $\mathcal{I}_u(x) \cap \mathcal{V}(x) = x$, où $\mathcal{I}_u(x)$ représente l'ensemble des états indiscernables de x_0 avec l'entrée u . Le système (A.11) est faiblement observable si et seulement s'il est faiblement observable pour tout $x \in \mathcal{X}$.*

Si le système (A.11) est localement observable et faiblement observable, alors il est **localement faiblement observable**. Cette notion est particulièrement intéressante et permet de proposer une caractérisation axée sur des conditions de rang. Une telle condition repose sur la notion d'espace d'observabilité, qui correspond à l'espace de l'ensemble des états observables.

Définition A.20 (Espace d'observabilité [10]) *L'espace d'observabilité du système (A.11) est défini comme le plus petit espace vectoriel réel (noté $\mathcal{O}(h)$) de fonctions \mathcal{C}^∞ contenant les composantes de h et fermé par la dérivation de Lie le long de $f_u := f(\cdot, u)$ pour n'importe quelle constante $u \in \mathbb{R}^m$ (telle que pour tout $\phi \in \mathcal{O}(h)$, $L_{f_u}\phi \in \mathcal{O}(h)$, où $L_{f_u}\phi = \frac{\partial \phi}{\partial x} f(x, u)$).*

Définition A.21 (Condition de rang pour l'observabilité [10]) *La condition de rang pour l'observabilité d'un système (A.11) est :*

$$\forall x \in \mathcal{X}, \dim d\mathcal{O}(h)|_x = n, \quad (\text{A.13})$$

où $d\mathcal{O}(h)|_x = \text{span}\{d\phi(x); \phi \in \mathcal{O}(h)\}$ est appelée la co-distribution d'observabilité.

Nous pouvons donc donner le théorème suivant :

Théorème A.1 [66] *Un système (A.11) satisfaisant la condition de rang pour l'observabilité (A.13), est localement faiblement observable. De façon équivalente un système (A.11) localement faiblement observable satisfait la condition de rang pour l'observabilité dans un sous-ensemble dense et ouvert de \mathcal{X} .*

Exemple : Considérons le système suivant

$$\begin{cases} \dot{x}_1 &= x_2 + x_1x_2 \\ \dot{x}_2 &= -x_1x_2 + u \\ y &= x_1. \end{cases} \quad (\text{A.14})$$

Dans ce système nous avons $h(x) = x_1$ et $f(x) = (x_2 + x_1x_2, -x_1x_2 + u)^T$. Nous avons donc $L_f h = x_2 + x_1x_2$, $dh = dx_1$ et $dL_f h = dx_2 + x_1dx_2 + dx_1x_2$. Ainsi, nous avons $d\mathcal{O}(h) = \text{span}\{dx_1, dx_2\}$ avec $\dim d\mathcal{O}(h) = 2$. Ce système est donc localement faiblement observable

A.3 CROCHET DE LIE

Définition A.22 ([Crochet de Lie]) *Le **crochet de Lie** (ou commutateur) défini par :*

$$[g_1, g_2] = \left(\frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2 \right),$$

permet de calculer la condition de commutativité de deux flots $\Phi_{g_1}^t$ et $\Phi_{g_2}^s$.

Théorème A.2 *Soient g_1 et g_2 des champs de vecteurs C^∞ complets, définis sur \mathcal{X} (par exemple \mathbb{R}^n). Alors :*

$$\forall t, \forall s, \quad \Phi_{g_1}^t \circ \Phi_{g_2}^s = \Phi_{g_2}^s \circ \Phi_{g_1}^t \Leftrightarrow [g_1, g_2] = 0.$$

Le crochet de Lie a différentes propriétés qui sont rappelées ci-dessous :

1. Le crochet de Lie est bilinéaire sur \mathbb{R} :

$$\begin{aligned} [\alpha_1 g_1 + g_2, g] &= \alpha_1 [g_1, g] + [g_2, g] \\ [g, \alpha_1 g_1 + g_2] &= \alpha_1 [g, g_1] + [g, g_2] \end{aligned}$$

2. Le crochet de Lie est anticommutatif :

$$[f, g] = -[g, f]$$

3. Le crochet de Lie satisfait l'identité de Jacobi :

$$[f, [g, h]] + [g, [h, f]] + [h, [f, g]] = 0$$

4. flot (commutativité) :

$$\forall t, \forall s, \quad \Phi_{g_1}^t \circ \Phi_{g_2}^s = \Phi_{g_2}^s \circ \Phi_{g_1}^t \Leftrightarrow [g_1, g_2] = 0.$$

En automatique, la non-commutativité des champs a une application très importante puisqu'elle permet de caractériser l'atteignabilité (version locale de la commandabilité) d'un système commandé du type

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2.$$

Cette caractérisation est aussi valable de façon générale pour des systèmes de la forme

$$\dot{x} = f(x) + \sum_{i=1}^p g_i(x)u_i, x \in \mathbb{R}^n$$

Dans ce cas il faut que

$$\text{rang}(\mathcal{A}\{f, g_1, g_2, \dots, g_p\}) = n,$$

où $\mathcal{A}\{f, g_1, g_2, \dots, g_p\}$ est l'algèbre de Lie engendrée par les champs de vecteurs $\{f, g_1, g_2, \dots, g_p\}$.

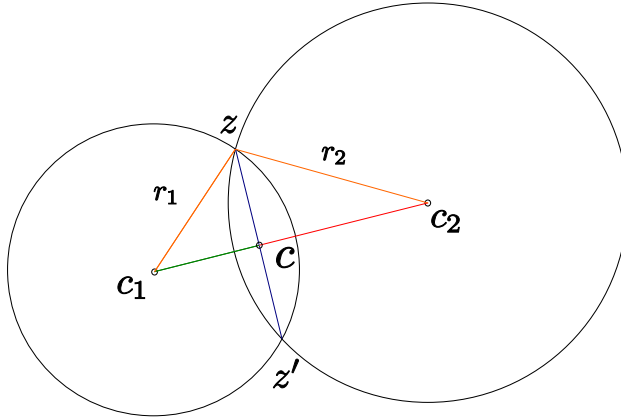
f est le champ de dérive.

Il est à noter que les modèles cinématiques que l'on va rencontrer ici sont sans dérive, c'est-à-dire que $f = 0$.

Remarque A.1 *Le crochet de Lie de deux champs de vecteurs g_1, g_2 est un nouveau champ de vecteurs $[g_1, g_2]$, défini par*

$$[g_1, g_2](x) = \left(\frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2 \right)(x). \quad (\text{A.15})$$

A.4 INTERSECTION DE DEUX CERCLES



En utilisant une notation complexe, les deux cercles de centre c_ℓ et de rayon r_ℓ , $\ell = 1, 2$ s'intersectent en un point z donné par :

$$z = c_1 + c_{12} \left(f(r_1, r_2) \pm i \sqrt{r_1^2 - f^2(r_1, r_2)} \right), \quad (\text{A.16})$$

$$f(r_1, r_2) = \frac{r_1^2 - r_2^2 + d_{12}^2}{2d_{12}}, \quad (\text{A.17})$$

où c_{12}, d_{12} sont données par :

$$c_{12} = \frac{c_2 - c_1}{d_{12}}, \quad d_{12} = |c_2 - c_1|. \quad (\text{A.18})$$

Remarque A.2 Les deux cercles de centre c_ℓ et de rayon r_ℓ , $\ell = 1, 2$ s'intersectent si et seulement si :

$$r_1 + r_2 > |c_1 - c_2|, \quad (\text{A.19})$$

$$|c_1 - c_2| > |r_1 - r_2|. \quad (\text{A.20})$$

La première équation est l'inégalité triangulaire. La seconde équation permet de s'affranchir du cas où l'un des deux cercles est inclus dans l'autre. Comme notre étude de localisabilité implique l'étude de deux cercles de centre c_{A_ℓ} et de rayon $|z_{r,\ell}|$, $\ell = 1, 2$ les conditions précédentes peuvent être réécrites sous la forme suivante :

$$|c_{A_1} - z| + |c_{A_2} - z| > |c_{A_1} - c_{A_2}|, \quad (\text{A.21})$$

$$|c_{A_1} - c_{A_2}| > ||c_{A_1} - z| - |c_{A_2} - z||. \quad (\text{A.22})$$

Si les deux points d'intérêt ne coïncident pas, alors $|z_{r,1}|$ et $|z_{r,2}|$ ne peuvent pas être simultanément nuls et donc les conditions précédentes sont satisfaites. Les deux cercles ont une intersection non vide.

Pour construire nos estimateurs d'état (et de commande), nous avons parfois besoin de construire \dot{z} et \ddot{z} :

$$\dot{z} = c_{12} \left(\dot{f} \pm i \frac{r_1 \dot{r}_1 - f \dot{f}}{\sqrt{r_1^2 - f^2}} \right), \quad (\text{A.23})$$

$$\ddot{z} = c_{12} \left(\ddot{f} \pm i \frac{(r_1^2 - f^2)(r_1 \ddot{r}_1 - f \ddot{f}) - (r_1 \dot{r}_1 - r_2 \dot{r}_2)^2}{(r_1^2 - f^2)^{\frac{3}{2}}} \right). \quad (\text{A.24})$$

A.5 ÉTUDE DE LA PLATITUDE À L'AIDE DE MAPLE

A.5.1 Platitudes d'un robot de type (2, 0)

```

# Inclusion des packages nécessaires
restart;
with(DifferentialAlgebra);
# Définition d'un anneau différentiel R
R := DifferentialRing(derivations=[t], blocks = [ [a, b, u,
w, x, y],[z, zc, i]]);
# Définition du modèle cinématique du système (en
considérant que \Theta = a + ib)
sys := [
x[t] = a * v,
y[t] = b * v,
a[t] = -b*w,
b[t] = a*w,
i^2+1 = 0,
a^2 + b^2 = 1,
z = x + i*y,
zc = x - i*y,
v<>0
];
Ids := RosenfeldGroebner(sys, R);
id := Ids[1];
# Les équations de platitude sont données par :
Equations(id, solved);
# Il faut ensuite appliquer la fonction racine sur v^2
pour trouver la bonne expression de v

```

A.5.2 Platitude d'un robot de type (3,0)

```

# Inclusion des packages nécessaires
restart;
with(DifferentialAlgebra);
# Définition d'un anneau différentiel R, on va utiliser z
# et \Theta comme sortie plates.
R := DifferentialRing(derivations=[t], blocks = [ [u1, u2,
w, x, y, a, b], [z, zc, T, Tc, i] ]):
# Définition du modèle cinématique du système (en
# considérant que \Theta = a + ib)
sys := [
x[t] = a*u1 - b*u2,
y[t] = a*u2 + b*u1,
a[t] = -b*w,
b[t] = a*w,
i^2+1 = 0,
a^2 + b^2 = 1,
z = x + i*y,
zc = x - i*y,
T = a + i*b,
Tc = a - i*b,
u<>0
];
Ids := RosenfeldGroebner(sys, R);
id := Ids[1];
Equations(id, solved);

```

A.5.3 Platitude d'un robot de type (2,1)

```

# Inclusion des packages nécessaires
restart;
with(DifferentialAlgebra);
# Définition d'un anneau différentiel R, on va utiliser z
# et B comme sortie plates.
R := DifferentialRing(derivations=[t], blocks = [[a, b, c, d, x
, y, u, w1, w2], [z, zc, B, Bc, i] ]):
# Définition du modèle cinématique du système (en
# considérant que \Theta = a + ib, B=c + id)
sys := [
x[t] = -u*a*d - u*b*c,
y[t] = u*a*c - u*b*d,
a[t] = -b*w1,
b[t] = a*w1,
c[t] = -d*w2,
d[t] = c*w2,
i^2+1 = 0,
a^2 + b^2 = 1,
c^2 + d^2 = 1,
z = x + i*y,
zc = x - i*y,
B = c + i*d,
Bc = c - i*d,
u<>0,
w1 <> 0,
w2 <> 0
];
Ids := RosenfeldGroebner(sys, R);
id := Ids[1];
Equations(id, solved);

```

A.5.4 Platitude d'un robot de type (1,1)

```

# Inclusion des packages nécessaires
restart;
with(DifferentialAlgebra);

```

```
# Définition d'un anneau différentiel R, on va utiliser z
comme sortie plate.
R := DifferentialRing(derivations=[t],blocks = [[L,u,w,a,
b,c,d,x,y],[z,zc,i]]);
# Définition du modèle cinématique du système (en
considérant que  $\Theta = a + ib$ ,  $B=c + id$ )
sys := [
x[t] == -L*u*b*d,
y[t] = a*L*u*d,
a[t] = -u*b*c,
b[t] = u*a*c,
c[t] = -w*d,
d[t] = c*w,
i^2+1 = 0,
a^2 + b^2 = 1,
c^2 + d^2 = 1,
z = x + i*y,
zc = x - i*y,
u<>0,
L<>0,
w<>0
];
Ids := RosenfeldGroebner(sys, R );
id := Ids[1];
Equations(id, solved);
```

BIBLIOGRAPHIE

- [1] Y. Abbasi-Yadkori, J. Modayil, and C. Szepesvari. Extending rapidly-exploring random trees for asymptotically optimal anytime motion planning. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 127–132. IEEE, 2010. (Cité page 27.)
- [2] R.S. Anderssen, F. De Hoog, and M. Hegland. A stable finite difference ansatz for higher order differentiation of non-exact data. *Bull. Austral. Math. Soc.*, 58 :223–232, 1998. (Cité page 64.)
- [3] R.C. Arkin. Motor schema-based mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 264–271, 1987. (Cité page 27.)
- [4] Abdul Bais, Robert Sablatnig, and Jason Gu. Single landmark based self-localization of mobile robots. *Canadian Conference on Computer and Robot Vision*, page 67, 2006. (Cité page 24.)
- [5] P. Barak. Smoothing and differentiation by an adaptive-degree polynomial filter. *Anal. Chem.*, 67 :2758–2762, 1995. (Cité page 63.)
- [6] J.P. Barbot, M. Fliess, and T. Floquet. An algebraic framework for the design of nonlinear observers with unknown inputs. In *46th IEEE Conference on Decision and Control*, New-Orleans, USA, 2007. (Cité page 63.)
- [7] H. Bay, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. *Computer Vision–ECCV*, pages 404–417, 2006. (Cité page 23.)
- [8] F.A.W. Belo, P. Salaris, and A. Bicchi. Three known landmarks are enough for solving planar bearing slam and fully reconstruct unknown inputs. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2539–2545. IEEE, 2010. (Cité page 25.)
- [9] S.J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18(3) :147–150, 1984. (Cité page 51.)
- [10] G. Besançon. Nonlinear observers and applications. *Lecture Notes in Control and Information Sciences*, 363, 2007. (Cité pages 160 et 161.)
- [11] A. Bicchi, D. Prattichizzo, A. Marigo, and A. Balestrino. On the observability of mobile vehicles localization. In *IEEE Mediterranean Conference on Control and Systems*, 1998. (Cité page 25.)
- [12] J. Biswas and M. Veloso. Wifi localization and navigation for autonomous indoor mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 4379–4384, 2010. (Cité page 24.)
- [13] P. Bonnifait and G. Garcia. Design and experimental validation of an odometric and goniometric localization system for outdoor robot vehicles. *Robotics and Automation, IEEE Transactions on*, 14(4) :541–548, 1998. (Cité page 25.)

- [14] J. Borenstein, HR Everett, and L. Feng. Where am I? Sensors and methods for mobile robot positioning. *University of Michigan*, 1996. (Cité page 22.)
- [15] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In *1990 IEEE International Conference on Robotics and Automation*,, pages 572–577. IEEE, 1990. (Cité page 27.)
- [16] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*,, 7(3) :278–288, 1991. (Cité page 27.)
- [17] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1) :14–23, 1986. (Cité page 27.)
- [18] R.H. Brown, S.C. Schneider, and M.G. Mulligan. Analysis of algorithms for velocity estimation from discrete position versus time data. *IEEE Trans. on industrial electronics*, 39(1) :11–19, 1992. (Cité page 64.)
- [19] G. Campion, G. Bastin, and B. Dandrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1) :47–62, 1996. (Cité pages 25, 31, 33 et 48.)
- [20] A. Charalampidis and G. Papavassilopoulos. Computationally efficient kalman filtering for a class of nonlinear systems. *IEEE Transactions on Automatic Control*, (99) :1–1, 2011. (Cité page 23.)
- [21] C.K. Chen and J.H. Lee. Design of high-order digital differentiators using l_1 error criteria. *IEEE Trans. Circuits Syst. II*, 42(4) :287–291, Avril 1995. (Cité page 63.)
- [22] S. Chen. Kalman filter for robot vision : a survey. *IEEE Transactions on Industrial Electronics*, (99) :1–1, 2012. (Cité page 23.)
- [23] J. Cheng, Y.C. Hon, and Y.B. Wang. A numerical method for the discontinuous solutions of abel integral equations. *Inverse Problems and Spectral Theory*, 348 :233–243, 2004. (Cité page 63.)
- [24] Y. Chitour. Time-varying high-gain observers for numerical differentiation. *IEEE Trans. Automat. Control*, 47 :1565–1569, 2002. (Cité page 63.)
- [25] H. Cho and S.W. Kim. Mobile robot localization using biased chirp-spread-spectrum ranging. *IEEE Transactions on Industrial Electronics*, 57(8) :2826–2835, 2010. (Cité page 23.)
- [26] T.H. Collett, B.A. MacDonald, and B.P. Gerkey. Player 2.0 : Toward a practical robot programming framework. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005)*, 2005. (Cité page 145.)
- [27] Giuseppe Conte, Anna Maria Perdon, and Claude Moog, H. The Differential Field Associated to a General Analytic Nonlinear Dynamical System. *IEEE Transactions on Automatic Control*, 38(7) :1120–1124, 1993. :hal-00409998 :hal-00409998 :hal-00409998 :hal-00409998
- [28] F. Conticelli, A. Bicchi, and A. Balestrino. Observability and nonlinear observers for mobile robot localization. In *IFAC International Symposium on Robot Control*, 2000. (Cité page 25.)
- [29] J. Correa and A. Soto. Active visual perception for mobile robot localization. *Journal of Intelligent & Robotic Systems*, 58(3) :339–354, 2010. (Cité page 23.)

- [30] J. Cullum. Numerical differentiation and regularization. *SIAM J. Numer. Anal.*, 8 :254–265, 1971. (Cité page 63.)
- [31] N.X. Dao, B.J. You, and S.R. Oh. Visual navigation for indoor mobile robots using a single camera. In *IEEE International Conference on Intelligent Robots and Systems*, 2005. (Cité page 24.)
- [32] M. Defoort, A. Kokosy, T. Floquet, W. Perruquetti, and J. Palos. Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges : A distributed receding horizon approach. *Robotics and Autonomous Systems*, 57(11) :1094–1106, 2009. (Cité pages 126, 127 et 130.)
- [33] Michael Defoort. *Contributions à la planification et à la commande pour les robots mobiles coopératifs*. PhD thesis, Ecole Centrale de Lille, 10 2007. :these2007 :these2007 :these2007 :these2007
- [34] G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2) :237–267, 2002. (Cité page 23.)
- [35] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1) :269–271, 1959. (Cité page 26.)
- [36] S. Diop. Differential-algebraic decision methods and some applications to system theory. *Theoretical computer science*, 98(1) :137–161, 1992. (Cité page 51.)
- [37] S. Diop, J.W. Grizzle, and F. Chaplais. On numerical differentiation algorithms for nonlinear estimation. In *Proc. 39th IEEE Conf. Decision Control*, Sidney, 2000. (Cité page 63.)
- [38] S. Diop, J.W. Grizzle, P.E. Moraal, and A. Stefanopoulou. Interpolation and numerical differentiation for observer design. In *Proc. Amer. Control Conf.*, pages 1329–1333, Baltimore, 1994. (Cité page 63.)
- [39] F.F. Dou, C.L. Fu, and Y.J. Ma. A wavelet-galerkin method for high order numerical differentiation. *Appl. Math. Comput.*, 215 :3702–3712, 2010. (Cité page 63.)
- [40] G. Dudek and M. Jenkin. *Computational principles of mobile robotics*. Cambridge Univ Pr, 2010. (Cité page 25.)
- [41] S. Even. *Graph algorithms*. Cambridge Univ Pr, 2011. (Cité page 27.)
- [42] M. Fliess. Automatique et corps différentiels. In *Forum Math*, volume 1, pages 241–242, 1989. (Cité page 51.)
- [43] M. Fliess. Analyse non standard du bruit. *C.R. Acad. Sci. Paris Ser. I*, 342 :797–802, 2006. (Cité page 63.)
- [44] M. Fliess. Critique du rapport signal à bruit en communications numériques - Questioning the signal to noise ratio in digital communications. in : *International Conference in Honor of Claude Lobry, Revue africaine d’informatique et de Mathématiques appliquées*, 9 :419–429, 2008. (Cité page 63.)
- [45] M. Fliess, C. Join, M. Mboup, and H. Sira-Ramírez. Compression différentielle de transitoires bruités. *C.R. Acad. Sci. Paris Ser. I*, 339 :821–826, 2004. (Cité page 63.)
- [46] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Flatness and defect of nonlinear systems : introductory theory and examples. *International journal of control*, 61(6) :1327–1361, 1995. (Cité page 133.)

- [47] M. Fliess, M. Mboup, H. Mounier, and H. Sira-Ramírez. Questioning some paradigms of signal processing via concrete examples. In G. Silva-Navarro H. Sira-Ramírez, editor, *in Algebraic Methods in Flatness, Signal Processing and State Estimation*, pages 1–21. Editorial Lagares, 2003. (Cité page 63.)
- [48] M. Fliess and H. Sira-Ramírez. An algebraic framework for linear identification. *ESAIM Control Optim. Calc. Variat.*, 9 :151–168, 2003. (Cité page 63.)
- [49] M. Fliess and H. Sira-Ramírez. Control via state estimations of some nonlinear systems. In *Proc. Symp. Nonlinear Control Systems (NOLCOS 2004)*, Stuttgart, 2004. (Cité page 63.)
- [50] M. Fliess and H. Sira-Ramírez. Reconstructeurs d'état. *C.R. Acad. Sci. Paris Ser. I*, 338 :91–96, 2004. (Cité page 63.)
- [51] M. Fliess and H. Sira-Ramírez. Closed-loop parametric identification for continuous-time linear systems via new algebraic techniques. In H. Garnier & L. Wang, editor, *Identification of Continuous-time Models from Sampled Data*, pages 363–391. Springer, Berlin, 2007. (Cité page 63.)
- [52] J.M. Font-Llagunes and J.A. Batlle. Consistent triangulation for mobile robot localization using discontinuous angular measurements. *Robotics and Autonomous Systems*, 57(9) :931–942, 2009. (Cité page 24.)
- [53] K. Forsman. Applications of gröbner bases to nonlinear systems. In *In Proceedings of first european control conference*. Citeseer, 1991. (Cité page 51.)
- [54] C.L. Fu, X.L. Feng, and Z. Qian. Wavelets and high order numerical differentiation. *Appl. Math. Model.*, 34 :3008–3021, 2010. (Cité page 63.)
- [55] T. Gabriel. *Contributions théoriques et algorithmiques à l'étude des équations différentielles algébriques. Approche par le calcul formel **. PhD thesis, Institut national Polytechnique de Grenoble, 1997. (Cité page 51.)
- [56] François Gaillard. *Approche cognitive pour la planification de trajectoire sous contraintes*. PhD thesis, Université Lille 1, Février 2012. (Cité page 133.)
- [57] G. Giralt, R. Chatila, and M. Vaisset. An integrated navigation and motion control system for autonomous multisensory mobile robots. In *First International Symposium on Robotics Research*, pages 191–214. ed. M Brady and R. Paul, 1984. (Cité page 49.)
- [58] R. Gorenflo and S. Vessella. Abel integral equations : Analysis and applications. *Lecture Notes in Mathematics*, 1461 :233–243, 1991. (Cité page 63.)
- [59] P.A. Gorry. General least-squares smoothing and differentiation by the convolution (savitzky-golay) method. *Anal. Chem.*, 62 :570–573, 1990. (Cité pages 63 et 64.)
- [60] M. Hanke and O. Scherzer. Error analysis of an equation error method for the identification of of the diffusion coefficient in a quasi-linear parabolic differential equation. *SIAM J. Appl. Math.*, 59(3) :1012–1027, 1999. (Cité page 63.)
- [61] M. Hanke and O. Scherzer. Inverse problems light : numerical differentiation. *Am. Math. Monthly*, 108 :512–521, 2001. (Cité page 63.)

- [62] D.N. Hào, A. Schneider, and H.J. Reinhardt. Regularization of a non-characteristic cauchy problem for a parabolic equation. *Inverse Probl.*, 11(6) :1247–1264, 1995. (Cité page 63.)
- [63] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50, 1988. (Cité page 23.)
- [64] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2) :100–107, 1968. (Cité page 26.)
- [65] D. Herceg and L. Cvetković. On a numerical differentiation. *SIAM J. NUMER. ANAL.*, 23(3), June 1986. (Cité page 64.)
- [66] R. Hermann and A.J. Krener. Nonlinear controllability and observability. *IEEE Trans. Automat. Control*, 22(5) :728–740, 1977. (Cité page 162.)
- [67] E. Hubert. *Etude algébrique et algorithmique des singularités des équations différentielles implicites*. PhD thesis, Institut national Polytechnique de Grenoble, 1997. (Cité page 51.)
- [68] S. Ibrir. Linear time-derivatives trackers. *Automatica*, 40 :397–405, 2004. (Cité page 63.)
- [69] Gijeong Jang, Sungho Kim, Jeongho Kim, and Inso Kweon. Metric localization using a single artificial landmark for indoor mobile robots. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1992 – 1997, 2005. (Cité page 24.)
- [70] I. Kamon and E. Rivlin. Sensory-based motion planning with global proofs. *IEEE Transactions on Robotics and Automation*, 13(6) :814–822, 1997. (Cité page 27.)
- [71] J.K. Kauppinen, D.J. Moffatt, H.H. Mantsch, and D.G. Cameron. Fourier transforms in the computation of self-deconvoluted and first-order derivative spectra of overlapped band contours. *Anal. Chem.*, 53 :1454–1457, 1981. (Cité page 63.)
- [72] T. Keviczky, F. Borrelli, and G.J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12) :2105–2115, 2006. (Cité page 127.)
- [73] I.R. Khan and R. Ohba. New finite difference formulas for numerical differentiation. *J. Comput. Appl. Math.*, 126 :269–276, 2000. (Cité page 63.)
- [74] D.E. Knuth. *The art of computer programming : Generating all trees : history of combinatorial generation*, volume 4. addison-Wesley, 2006. (Cité page 27.)
- [75] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *1991 IEEE International Conference on Robotics and Automation*, , pages 1398–1404. IEEE, 1991. (Cité page 27.)
- [76] Y. Kuwata, A. Richards, T. Schouwenaars, and J.P. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4) :627–641, 2007. (Cité page 127.)
- [77] C. Lanczos. *Applied Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1956. (Cité page 63.)
- [78] S.M. LaValle and J.J. Kuffner Jr. Rapidly-exploring random trees : Progress and prospects. 2000. (Cité page 27.)

- [79] C. Lawrence, J.L. Zhou, and A.L. Tits. User's guide for CFSQP version 2.5 : AC code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints. *Institute for Systems Research TR*, 94 :16r1, 94. (Cité page 133.)
- [80] T. Lemaire, S. Lacroix, and J. Sola. A practical 3D bearing-only SLAM algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2449–2454, 2005. (Cité page 24.)
- [81] A.K.M. Leung, F.T. Chau, and J.B. Gao. Wavelet transform : A method for derivative calculation in analytical chemistry. *Anal. Chem.*, 70 :5222–5229, 1998. (Cité page 63.)
- [82] A. Levant. Higher-order sliding modes, differentiation and output-feedback control. *Int. J. Control*, 76 :924–941, 2003. (Cité page 63.)
- [83] D.Y. Liu. *Analyse d'Erreurs d'Estimateurs des Dérivées de Signaux Bruités et Applications*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I, 2011. (Cité page 67.)
- [84] D.Y. Liu, O. Gibaru, and W. Perruquetti. Error analysis for a class of numerical differentiator : application to state observation. In *48th IEEE Conf. on Decision and Control*, Shanghai, China, 2009. (Cité page 63.)
- [85] D.Y. Liu, O. Gibaru, and W. Perruquetti. Convergence rate of the causal jacobi derivative estimator. *Lecture Notes in Computer Science*, 2011. (Cité pages 63 et 67.)
- [86] D.Y. Liu, O. Gibaru, and W. Perruquetti. Differentiation by integration with jacobi polynomials. *J. Comput. Appl. Math.*, 235 :3015–3032, 2011. (Cité page 63.)
- [87] D.Y. Liu, O. Gibaru, and W. Perruquetti. Error analysis of jacobi derivative estimators for noisy signals. *Numerical Algorithms*, 2011. (Cité pages 63 et 67.)
- [88] D.Y. Liu, O. Gibaru, W. Perruquetti, M. Fliess, and M. Mboup. An error analysis in the algebraic estimation of a noisy sinusoidal signal. In *16th Mediterranean Conf. on Control and Automation (MED'08)*, Ajaccio, France, 2008. (Cité page 63.)
- [89] L. Ljung and T. Glad. Parametrization of nonlinear model structures as linear regressions. In *11th IFAC world congress (Tallin, Estonia)*, 1990. (Cité page 51.)
- [90] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004. (Cité page 23.)
- [91] T. Lozano-Perez. Spatial planning : A configuration space approach. *IEEE Transactions on Computers*, 100(2) :108–120, 1983. (Cité page 131.)
- [92] V.J. Lumelsky, S. Mukhopadhyay, and K. Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, 6(4) :462–472, 1990. (Cité page 27.)
- [93] V.J. Lumelsky and A.A. Stepanov. Path-planning strategies for a point mobile automaton moving among unknown obstacles of arbitrary shape. *Algorithmica*, 2(1) :403–430, 1987. (Cité page 27.)
- [94] E. Magid and E. Rivlin. Cautiousbug : A competitive algorithm for sensory-based robot navigation. In *2004 IEEE/RSJ International Conference on*

- Intelligent Robots and Systems*, volume 3, pages 2757–2762. IEEE, 2004. (Cité page 27.)
- [95] G. Margaria. *Applications of differential algebra to the structural identifiability of non linear models*. PhD thesis, Department of Mathematics, Politecnico di Torino, 2000. (Cité page 51.)
- [96] P. Martin and P. Rouchon. Feedback linearization and driftless systems. *Mathematics of Control, Signals, and Systems (MCSS)*, 7(3) :235–254, 1994. (Cité page 59.)
- [97] P. Martin and P. Rouchon. Any (controllable) driftless system with 3 inputs and 5 states is flat. *Systems & control letters*, 25(3) :167–173, 1995. (Cité page 59.)
- [98] P. Martin and P. Rouchon. Any (controllable) driftless system with m inputs and $m+2$ states is flat. In *1995 Proceedings of the 34th IEEE Conference on Decision and Control*, volume 3, pages 2886–2891. IEEE, 1995. (Cité page 59.)
- [99] A. Martinelli and R. Siegwart. Observability analysis for mobile robot localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1471–1476, 2005. (Cité page 25.)
- [100] M. Mboup. Parameter estimation for signals described by differential equations. *Appl. Anal.*, 88 :29–52, 2009. (Cité page 63.)
- [101] M. Mboup, C. Join, and M. Fliess. A revised look at numerical differentiation with an application to nonlinear feedback control. In *15th Mediterranean Conf. on Control and Automation (MED'07)*, Athenes, Greece, 2007. (Cité pages 63, 64 et 66.)
- [102] M. Mboup, C. Join, and M. Fliess. Numerical differentiation with annihilators in noisy environment. *Numerical Algorithms*, 50(4) :439–467, 2009. (Cité pages 63, 64, 66 et 67.)
- [103] Henry Mintzberg. The structuring of organizations : A synthesis of the research. *University of Illinois at Urbana-Champaign's Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship*, 1979. (Cité page 26.)
- [104] H. Mounier. *Propriétés structurelles des systèmes linéaires à retards : aspects théoriques et pratiques*. PhD thesis, Université Paris Sud, Orsay, 1995. (Cité page 51.)
- [105] D.A. Murio. *The Mollification Method and the Numerical Solution of Ill-Posed Problems*. John Wiley & Sons Inc., 1993. (Cité page 63.)
- [106] D.A. Murio, C.E. Mejía, and S. Zhan. Discrete mollification and automatic numerical differentiation. *Comput. Math. Appl.*, 35 :1–16, 1998. (Cité page 63.)
- [107] G. Nakamura, S. Wang, and Y. Wang. Numerical differentiation for the second order derivatives of functions of two variables. *J. Comput. Appl. Math.*, 212 :341–358, 2008. (Cité page 63.)
- [108] J.I. Neimark and N.A. Fufaev. *Dynamics of nonholonomic systems*, volume 33. Amer Mathematical Society, 1972. (Cité page 33.)
- [109] A. Neves, M. Mboup, and M. Fliess. An algebraic receiver for full response cpm demodulation. In *VI International telecommunications symposium (ITS2006)*, Fortaleza, Ceara, Brazil, 2006. (Cité page 63.)

- [110] A. Neves, M.D. Miranda, and M. Mboup. Algebraic parameter estimation of damped exponentials. In *Proc. 15th Europ. Signal Processing Conf. - EUSIPCO 2007*, Poznań, 2007. (Cité page 63.)
- [111] L. Nie, S.G. Wu, X.Q. Lin, L.Z. Zheng, and L. Rui. Approximate derivative calculated by using continuous wavelet transform. *J. Chem. Inf. Comput. Sci.*, 42 :274–283, 2002. (Cité page 63.)
- [112] H. Nijmeijer and A.J. Van der Schaft. *Nonlinear dynamical control systems*. Springer, 1990. (Cité page 37.)
- [113] C. Noiret. Utilisation du calcul formel pour l'identifiabilité de modèles paramétriques et nouveaux algorithmes en estimation de paramètres. 2000. (Cité page 51.)
- [114] Y. Ogata, E. Spaho, K. Matsuo, L. Barolli, J. Arnedo-Moreno, and F. Xhafa. Jxta-overlay p2p platform and its application for robot control. In *2010 13th International Conference on Network-Based Information Systems (NBIS)*, pages 133–138, september 2010. (Cité page 145.)
- [115] Y. Ogata, E. Spaho, K. Matsuo, L. Barolli, and F. Xhafa. A knowledge sharing p2p system between robots using jxta-overlay. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2011 International Conference on*, pages 364–370, 30 2011-july 2 2011. (Cité page 145.)
- [116] F. Ollivier. *Le problème de l'identifiabilité structurelle globale : approche théorique, méthodes effectives et bornes de complexité*. PhD thesis, Ecole Polytechnique, 1990. (Cité page 51.)
- [117] J. Pearl. Heuristics : intelligent search strategies for computer problem solving. 1984. (Cité page 27.)
- [118] A. Péladan Germa. *Tests effectifs de nullité dans des extensions d'anneaux différentiels*. PhD thesis, Ecole Polytechnique, 1997. (Cité page 51.)
- [119] Z. Qian, C.L. Fu, and X.L. Feng. A modified method for high order numerical derivatives. *Appl. Math. Comput.*, 182 :1191–1200, 2006. (Cité page 63.)
- [120] Z. Qian, C.L. Fu, X.T. Xiong, and T. Wei. Fourier truncation method for high order numerical derivatives. *Appl. Math. Comput.*, 181 :940–948, 2006. (Cité page 63.)
- [121] R. Qu. A new approach to numerical differentiation and integration. *Math. Comput.*, 24(10) :55–68, 1996. (Cité page 63.)
- [122] C.M. Rader and L.B. Jackson. Approximating noncausal iir digital filters having arbitrary poles, including new hilbert transformer designs, via forward/backward block recursion. *IEEE Trans. Circuits Syst. I*, 53(12) :2779–2787, December 2006. (Cité page 63.)
- [123] AN Raghavan, H. Ananthapadmanaban, MS Sivamurugan, and B. Ravindran. Accurate mobile robot localization in indoor environments using bluetooth. In *IEEE International Conference on Robotics and Automation*, pages 4391–4396, 2010. (Cité page 24.)
- [124] K. Rahul and S.N. Bhattacharyya. One-sided finite-difference approximations suitable for use with richardson extrapolation. *J. Comput. Phys.*, 219 :13–20, 2006. (Cité page 63.)

- [125] A. RAKSANYI. *Utilisation du calcul formel pour l'étude des systèmes d'équations polynomiales (applications en modélisation)*. PhD thesis, Université Paris-Dauphine, 1986. (Cité page 51.)
- [126] A.G. Ramm and A.B. Smirnova. On stable numerical differentiation. *Math. Comput.*, 70 :1131–1153, July 2001. (Cité page 63.)
- [127] S.K. Rangarajana and S.P. Purushothaman. Lanczos' generalized derivative for higher orders. *J. Comput. Appl. Math.*, 177 :461–465, 2005. (Cité page 63.)
- [128] G.J. Reid, P. Lin, and A.D. Wittkopf. Differential elimination–completion algorithms for dae and pdae. *Studies in Applied Mathematics*, 106(1) :1–45, 2001. (Cité page 51.)
- [129] Jean-Pierre Richard, Hugues Mounier, Abdennebi Achour, Lotfi Belkoura, Selma Ben Attia, Michel Dambrine, Mekki Ksouri, Wilfrid Perruquetti, Joachim Rudolph, Salah Salhi, and Franck Woittennek. *Mathématiques pour l'ingénieur*. ATAN, Association Tunisienne d'Automatique et de Numérisation, 2009. :hal-00519555 :hal-00519555 :hal-00519555 :hal-00519555
- [130] P. Rudol, M. Wzorek, and P. Doherty. Vision-based pose estimation for autonomous indoor navigation of micro-scale unmanned aircraft systems. In *IEEE International Conference on Robotics and Automation*, pages 1913–1920, 2010. (Cité page 24.)
- [131] B. Sadik. The complexity of formal resolution of linear partial differential equations. *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 408–414, 1995. (Cité page 51.)
- [132] A. Savitzky and M.J.E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.*, 36 :1627–1638, 1964. (Cité pages 63 et 64.)
- [133] A. Sedoglavic. *Méthodes seminumériques en algèbre différentielle ; applications à l'étude des propriétés structurelles de systèmes différentiels algébriques en automatique*. PhD thesis, Ecole polytechnique, 2001. (Cité page 51.)
- [134] H. Sert, A. Kökösy, J. Palos, and W. Perruquetti. Mono landmark localization for an autonomous navigation of a cooperative mobile robot formation. *Intelligent Robotics and Applications*, pages 1278–1292, 2009. (Cité page 82.)
- [135] H. Sert, A. Kokosy, and W. Perruquetti. A single landmark based localization algorithm for non-holonomic mobile robots. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 293–298. IEEE, 2011. (Cité page 82.)
- [136] H. Sert, W. Perruquetti, A. Kokosy, X. Jin, J. Palos, et al. Localizability of unicycle mobile robots : an algebraic point of view. In *2012 IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012. (Cité page 94.)
- [137] X.G. Shao and C.X. Ma. A general approach to derivative calculation using waveletnext term transform. *Chemometrics and Intelligent Laboratory Systems*, 69 :157–165, 2003. (Cité page 63.)
- [138] X.G. Shao, C.Y. Pang, and Q.D. Su. A novel method to calculate the approximate derivative photoacoustic spectrum using continuous wavelet transform. *Fresenius' J. Anal. Chem.*, 367 :525–529, 2000. (Cité page 63.)

- [139] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994. (Cité page 23.)
- [140] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, and K. Schwehr. Recent progress in local and global traversability for planetary rovers. In *2000 IEEE International Conference on Robotics and Automation*, volume 2, pages 1194–1200. IEEE, 2000. (Cité page 27.)
- [141] A. Stentz. The focussed d^* algorithm for real-time replanning. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1652–1659. LAWRENCE ERLBAUM ASSOCIATES LTD, 1995. (Cité page 27.)
- [142] A. Stentz. Optimal and efficient path planning for partially known environments. *Intelligent Unmanned Ground Vehicles*, pages 203–220, 1997. (Cité page 27.)
- [143] J.R. Trapero, H. Sira-Ramírez, and V.F. Battle. An algebraic frequency estimator for a biased and noisy sinusoidal signal. *Signal Processing*, 87 :1188–1201, 2007. (Cité page 63.)
- [144] J.R. Trapero, H. Sira-Ramírez, and V.F. Battle. A fast on-line frequency estimator of lightly damped vibrations in flexible structures. *J. Sound Vibration*, 307 :365–378, 2007. (Cité page 63.)
- [145] J.R. Trapero, H. Sira-Ramírez, and V.F. Battle. On the algebraic identification of the frequencies, amplitudes and phases of two sinusoidal signals from their noisy sums. *Int. J. Control*, 81 :505–516, 2008. (Cité page 63.)
- [146] C.C. Tseng. Digital differentiator design using fractional delay filter and limit computation. *IEEE Trans. Circuits Syst. I*, 52(10) :2248–2259, October 2005. (Cité page 63.)
- [147] J.K. Uhlmann. Algorithms for multiple-target tracking. *American Scientist*, 80(2) :128–141, 1992. (Cité page 23.)
- [148] I. Ulrich and J. Borenstein. Vfh+ : Reliable obstacle avoidance for fast mobile robots. In *1998 IEEE International Conference on Robotics and Automation*, volume 2, pages 1572–1577. IEEE, 1998. (Cité page 27.)
- [149] T.T. Um, B. Kim, C. Suh, and F.C. Park. Tangent space rrt with lazy projection : An efficient planning algorithm for constrained motions. *Advances in Robot Kinematics : Motion in Man and Machine*, pages 251–260, 2010. (Cité page 27.)
- [150] Y. Wang, X. Jia, and J. Cheng. A numerical differentiation method and its application to reconstruction of discontinuity. *Inverse Probl.*, 18 :1461–1476, 2002. (Cité page 63.)
- [151] Z. Wang and J. Liu. Identification of the pollution source from one-dimensional parabolic equation models. *Appl. Math. Comput.*, 2008. (Cité page 63.)
- [152] Z. Wang and R. Wen. Numerical differentiation for high orders by an integration method. *J. Comput. Appl. Math.*, 234 :941–948, 2010. (Cité page 63.)
- [153] F.W. Warner. *Foundations of differentiable manifolds and Lie groups*, volume 94. Springer Verlag, 1971. (Cité page 37.)
- [154] T. Wei, Y.C. Hon, and Y. Wang. Reconstruction of numerical derivatives from scattered noisy data. *Inverse Probl.*, 21 :657–672, 2005. (Cité page 63.)

- [155] Lu Y. A perturbation method for numerical differentiation. *Appl. Math. Comput.*, 199 :368–374, 2008. (Cité page 63.)
- [156] W.S. You, B.J. Choi, B. Kim, H. Moon, J.C. Koo, W. Chung, and H.R. Choi. Global localization for a small mobile robot using magnetic patterns. In *IEEE International Conference on Robotics and Automation*,, pages 2618–2623, 2010. (Cité page 24.)

Titre De l'utilisation de l'algèbre différentielle pour la localisation et la navigation de robots mobiles autonomes

Résumé Ce travail étudie l'apport de l'algèbre différentielle à deux problématiques principales de la navigation autonome des robots mobiles à roues, la localisation et la navigation. La première problématique consiste à trouver la position du robot dans son environnement. Afin d'y apporter une solution, nous faisons ici l'hypothèse que le robot a accès à un certain nombre de points d'intérêt de l'espace de coordonnées connues. En fonction de leur nombre, il est possible ou non de localiser le robot. Cette notion, appelée localisabilité, est définie et étudiée dans le cadre algébrique. Nous montrons que ce cadre d'étude est plus intéressant que le cadre géométrique en ce sens que non seulement il permet l'étude de la localisabilité mais en plus il permet de construire des estimateurs d'états permettant de reconstruire la posture du robot. Nous proposons dans ce travail des estimateurs de la posture du robot ainsi que, dans certains cas, de ses vitesses linéaire et angulaire, pour cinq modèles de mesures différents (information sur l'azimut et/ou l'élévation d'un ou deux points d'intérêt) et quatre des cinq classes de robots mobiles à roues. Les performances de ces estimateurs sont comparées en simulation et en expérimentation à celles obtenues en utilisant un filtre de Kalman étendu. La deuxième problématique étudiée est celle de la navigation d'une flottille décentralisée de robots dans un environnement complexe connu que partiellement. Ce travail propose une architecture software pouvant être utilisée dans une large classe de problème et bénéficiant des avantages des approches discrètes et des approches continues. En effet, à haut niveau, un bloc stratégie spécifie l'objectif, les contraintes et leurs paramètres ainsi que la fonction coût utilisée, à bas niveau, une trajectoire est calculée en s'appuyant sur la propriété de platitude du système et en utilisant un algorithme d'optimisation non linéaire afin de minimiser la fonction coût en respectant l'objectif et les contraintes du problème. Cette minimisation est faite sur un horizon glissant de manière à pouvoir prendre en compte des modifications de l'environnement ou de la mission en cours de navigation. De plus, nous proposons un algorithme de coopération entre les robots, qui offre à chaque robot de la flottille la possibilité d'augmenter ses connaissances de l'environnement en utilisant ses voisins. Les résultats de simulations montrent qu'en coopérant, la flottille peut contenir plus de robots pour réaliser sa mission tout en limitant le temps de calcul nécessaire pour obtenir la trajectoire optimale pour chaque robot. Une expérimentation est également réalisée dans un environnement urbain à taille réduite en utilisant cinq robots mobiles, dont quatre autonomes pour démontrer l'efficacité de l'algorithme dans un environnement complexe et dynamique.

Mots-clés Robot mobile à roues, localisation, localisabilité, planification de trajectoire, stratégie, algèbre différentielle, différentiateur numérique, platitude, coopération, estimation de posture et de vitesse

Title The use of differential algebra for the localisation and autonomous navigation of wheeled mobile robots

Abstract This work investigates the contribution of differential algebra to two main issues of wheel mobile robotics, localization and navigation. The first issue is to be able to tell where the robot is in its environment. We assume that we have a number of landmarks in space whose coordinates are known in this area. Depending on the number of landmarks, it is possible or not to localize the robot. This notion of localizability is defined and studied in the algebraic framework. We show that this framework is more interesting than the geometric framework in the sense that it not only allows the study of localizability, but it also allows us to construct estimators states to reconstruct the posture of the robot. This study was conducted in five cases study for four of the five classes of wheeled mobile robots. The second problem studied is that of a robot decentralized swarm navigation in a complex environment. This work presents an architecture that can be used in a wide class of problems and enjoying the benefits of discrete approaches and continuous approaches. Indeed, high-level block strategy specifies the goal, constraints and parameters as well as the cost function, a low-level block is used to compute a trajectory that minimize the cost function in accordance with the objective and the problem constraints. This minimization is done on a sliding windows so it is possible to take changes in the environment or mission during navigation into account.

Keywords Wheeled mobile robot, localisation, localisability, path planning, decision, differential algebra, numerical differentiation, flat output, cooperation, posture and speed estimation