



Contribution à l'autonomie des robots : vers la garantie de performance en robotique mobile autonome par la gestion des ressources matérielles et logicielles

Lotfi Jaïem

► To cite this version:

Lotfi Jaïem. Contribution à l'autonomie des robots : vers la garantie de performance en robotique mobile autonome par la gestion des ressources matérielles et logicielles. Robotique [cs.RO]. Université Montpellier, 2016. Français. NNT : 2016MONTT338 . tel-01816954

HAL Id: tel-01816954

<https://theses.hal.science/tel-01816954>

Submitted on 15 Jun 2018

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

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

THÈSE

Pour obtenir le grade de
Docteur

Délivré par l'Université de Montpellier

Préparée au sein de l'école doctorale **I2S**
Et de l'unité de recherche **LIRMM**

Spécialité: **Génie Informatique et Automatique**

Présentée par **Lotfi JAÏEM**

**Contribution à l'autonomie des
robots : Vers la garantie de
performance en robotique mobile
autonome par la gestion des
ressources matérielles et logicielles**

Soutenue le 21/11/2016 devant le jury composé de

Mme Maryline CHETTO	Professeur	IRCCyN	Rapporteur
M. Simon LACROIX	Directeur de Recherche	LAAS	Rapporteur
M. Jacques MALENFANT	Professeur	LIP6	Examineur
M. Olivier PARODI	Docteur	DCNS	Examineur
M. René ZAPATA	Professeur	LIRMM	Président
M. Lionel LAPIERRE	MC-HdR	LIRMM	Co-Directeur de thèse
M. Didier CRESTANI	Professeur	LIRMM	Directeur de thèse
M. David ANDREU	MC-HdR	LIRMM	Invité



*... Lorsque je tends vers un but,
je me fais porter par l'espoir
et oublie toute prudence
Je n'évite pas les chemins escarpés,
et n'appréhende pas la chute dans un feu brûlant.
Qui n'aime pas gravir la montagne,
vivra éternellement au fond des vallées ...*

Extrait du poème "**La volonté de vivre**" d'Abou Kacem Chebbi, 1933.

Remerciements

La réalisation de cette thèse a été rendue possible grâce à la contribution, l'aide et le soutien de plusieurs personnes auxquelles je souhaiterais témoigner toute ma reconnaissance.

Je remercie chaleureusement toutes les personnes qui m'ont aidé pendant l'élaboration de ma thèse et notamment mes encadrants Monsieur Didier Crestani et Monsieur Lionel Lapierre. Je voudrais leur exprimer toute ma gratitude, pour les contacts humains que nous avons liés, leur intérêt, leur soutien, leur aide, leur grande disponibilité ainsi que leurs nombreux conseils durant la rédaction de ma thèse.

Je veux également remercier Robin Passama, le maître de ContrACT et Sebastien Druon pour leur disponibilité, leurs suggestions et leurs conseils qu'ils m'ont prodigué lors de la mise en œuvre de mes travaux.

Je souhaite remercier Madame Maryline Chetto et Monsieur Simon Lacroix d'avoir accepté la lourde tâche de relire cette thèse et d'en être rapporteurs. Je tiens à remercier Monsieur René Zapata pour avoir accepté d'être président de mon jury ainsi que pour notre collaboration dans plusieurs thématiques. Je remercie également les autres membres du jury Monsieur Jacques Malenfant, Monsieur Olivier Parodi et Monsieur David Andreu pour avoir accepté d'assister à la présentation de ce travail. Que l'ensemble des membres du jury soit remercié pour le temps passé à lire mon manuscrit et pour leurs commentaires précieux.

Mes remerciements vont ensuite à tous les membres de l'équipe EXPLORE pour le climat sympathique dans lequel ils m'ont permis de travailler. Les nombreuses discussions que j'ai pu avoir avec chacun m'ont beaucoup apporté. Merci donc à Karen Godary-Dejean, Jean Triboulet, Pascal Lepinay et Bruno Jouvencel.

Je remercie particulièrement mes camarades doctorants qui m'ont accompagné pendant ces trois ans de thèse. Ceux qui ont fini le combat Florent, Benoît, Adrien, Mohamed et Moussab et ceux qui luttent encore Yadpiroon, Rim, Silvain, Yohan et Gamal. Je les remercie tous pour leur encouragement, leur soutien et la bonne humeur.

Au terme de ce parcours, je remercie enfin celles et ceux qui me sont chers et que j'ai quelque peu délaissés ces derniers mois pour achever cette thèse. Leurs attentions et encouragements m'ont accompagnée tout au long de ces années. Je suis redevable à mes parents, Fathi et Fouzia et mes sœurs Hajer et Hazar pour leur soutien moral et matériel et leur confiance indéfectible dans mes choix. Enfin, j'ai une pensée toute particulière pour

mes grand-parents, Baba Hamed et Ya Chrifa qui n'ont pas cessé de me supporter depuis mon enfance.

Enfin, je remercie ma chère épouse Racha qui, malgré la distance, a su me soutenir, me supporter, m'encourager. . . pendant toute la durée de ma thèse et plus particulièrement durant les derniers mois de rédaction qui n'ont pas toujours été des plus agréables. Cette thèse et moi te devons beaucoup. Merci.

Table des matières

Introduction	1
1 De l'autonomie à la décision multicritère : État de l'art	7
1.1 Autonomie	8
1.1.1 Modes/niveaux d'autonomie	10
1.1.2 Autonomie comportementale	12
1.1.2.1 Concept de l'autonomie comportementale	13
1.1.2.2 De l'autonomie comportementale à la performance	15
1.2 Performance	16
1.2.1 Concept de performance	16
1.2.1.1 La performance : une entité complexe	17
1.2.1.2 Vues locale et globale	18
1.2.1.3 La performance : une vision plurielle	18
1.2.1.4 Les logiques d'évaluation de la performance	19
1.2.1.5 Évaluation de la performance	19
1.2.1.6 Indicateurs et inducteurs de performance	20
1.2.1.7 Pilotage de la performance	23
1.2.2 Performance en robotique	26
1.2.2.1 Robotique manufacturière	26
1.2.2.2 Robotique mobile	28
1.2.3 Conclusion	36
1.3 Décision multicritère	37
1.3.1 Mission et décision multicritère	38
1.3.2 Techniques de résolution	41
1.3.2.1 Méthodes probabilistes	41

1.3.2.2	Méthodes non probabilistes	43
1.4	Positionnement	46
2	Proposition d'une méthodologie de gestion de la performance des missions en robotique mobile	49
2.1	Axes de performance	49
2.1.1	Classification des axes de performance	50
2.1.1.1	Axes de performance principaux	51
2.1.1.2	Axes de performance secondaires	54
2.1.2	Caractéristiques de la performance	55
2.1.2.1	Notions liées à un axe de performance	55
2.1.2.2	Classification des performances	56
2.1.3	Marge de performance	58
2.2	Approche de gestion des ressources matérielles et logicielles proposée	60
2.2.1	Définitions	61
2.2.1.1	Mission et ressources : Définitions	61
2.2.1.2	Contraintes	62
2.2.1.3	Complexité	65
2.2.1.4	Principaux inducteurs de performance	66
2.2.1.5	Algorithme général de gestion des ressources guidée par la performance	69
2.2.2	Phase préliminaire (P1)	72
2.2.2.1	Identification des indicateurs et inducteurs de performance et des modèles d'estimation associés	72
2.2.2.2	Construction du Scénario de Mission Détaillé (SMD)	72
2.2.3	Estimation hors ligne de la performance (P2)	76
2.2.3.1	Formalisation sac à dos du problème d'allocation de ressources	76
2.2.3.2	Algorithme	77
2.2.4	Estimation et gestion en ligne de la performance (P3)	80
2.3	Conclusion approche de gestion des ressources	81
3	Contexte applicatif et expérimental	83
3.1	Plateforme expérimentale	83

3.1.1	Ressources matérielles	83
3.1.1.1	Plateforme mobile de base	84
3.1.1.2	Éléments ajoutés	85
3.1.2	Ressources logicielles	87
3.1.2.1	Architecture de contrôle	87
3.1.2.2	Modules fonctionnels	90
3.1.2.3	Option d'Implémentation de Tâches OIT retenues	99
3.2	Mission de patrouille (<i>patrolling</i>)	101
3.2.1	Description de la mission	101
3.2.2	Environnement	102
3.2.3	Contraintes de performance	103
3.3	Conclusion	104
4	Mise en œuvre de la phase statique de la méthodologie d'affectation des ressources	105
4.1	Méthodologie : Phase préliminaire P1	106
4.1.1	Axe sécurité	106
4.1.1.1	Graphe des indicateurs/inducteurs de performance de l'axe sécurité	106
4.1.1.2	Modèles d'estimation de la performance sécuritaire	108
4.1.2	Axe énergie	116
4.1.2.1	Graphe des indicateurs/inducteurs de performance de l'axe énergie	116
4.1.2.2	Modèles d'estimation de l'énergie	118
4.1.3	Axe durée	138
4.1.3.1	Graphe des indicateurs/inducteurs de performance de l'axe durée	138
4.1.3.2	Modèle d'estimation de la durée	140
4.1.4	Axe localisation	142
4.1.5	Axe stabilité	143
4.1.6	Identification du Scénario de Mission Détaillé (SMD)	145
4.1.6.1	SMD sans prise en compte des contraintes de performance	146
4.1.6.2	SMD avec prise en compte de l'axe sécurité	149

4.1.6.3	SMD avec prise en compte des contraintes de performance sécurité et localisation	151
4.1.7	Conclusion	153
4.2	Méthodologie : Phase hors ligne P2	153
4.2.1	Mise en œuvre de la phase P2 sur SMD sans performances boo- léennes et avec prise en compte de l'axe durée	154
4.2.2	Mise en œuvre de la phase P2 sur SMD avec prise en compte de l'axe sécurité (booléen) et de l'axe durée (continu)	155
4.2.3	Mise en œuvre de la phase P2 sur SMD avec prise en compte des axes booléens et continus	157
4.3	Conclusion	158
5	Mise en œuvre de la phase dynamique de la méthodologie	161
5.1	Simulation	163
5.1.1	Description de l'outil de simulation	163
5.1.2	Simulation	164
5.1.2.1	Génération d'événements	165
5.1.2.2	Déroulement de la mission simulée	165
5.2	Expérimentation	173
5.2.1	Conditions expérimentales	173
5.2.2	Déroulement de la mission	173
5.3	Conclusion	179
	Conclusion générale	181
	Annexe A	193
	Annexe B	197
	Publications de l'auteur	205
	Bibliographie	209

Table des figures

1.1	Liens entre les concepts d'autonomie, de décision, et de performance	8
1.2	Le tétraèdre des performances	17
1.3	Structure causes-effets [MBC01] : l'arborescence indiquant plusieurs ni- veaux de causalité qui lient un objet aux potentielles actions qui l'influencent.	21
1.4	Système d'indicateurs correspondant à la structure causes-effets de l'exemple de la Figure 1.3 [MBC01]	22
1.5	Modèle générique du pilotage de la performance proposé dans [Rav99] . . .	24
1.6	Modèle du centre de pilotage [Tah03]	24
1.7	Boucle de pilotage [Tah03]	25
1.8	Performance et tolérance à l'inattention	34
1.9	Schéma d'illustration de la mesure d'autonomie proposée par [Clo02] . . .	35
1.10	Optimisation basée mission	39
1.11	Un cycle de conception typique	39
1.12	Processus de décision selon Ullman (figure tirée de [Ull02])	40
1.13	Exemple d'arbre de décision (figure tirée de [Sel+06a])	42
1.14	Matrice de comparaison alternatives/critères	43
1.15	Illustration des hypothèses de classement de l'algorithme proposé dans [Ben+05]	45
2.1	Classification des axes de performance	51
2.2	Notions liées à la performance	56
2.3	Les classes de performance du système robotique $Perf_{SR}$ et performance contrainte $Perf_C$	57

2.4	Notion de marge de performance	59
2.5	Hypothèses de description d'une mission robotique	64
2.7	Exemple de compositions des alternatives d'implémentations AI.	64
2.6	Projection unidimensionnelle des objectifs, activités et contraintes.	65
2.8	Approche proposée de gestion des ressources guidée par la performance . .	71
2.9	Exemple de création d'un SMD	75
2.10	Formalisation sac à dos multicritère	78
2.11	Algorithme de recherche dichotomique proposé par [Ben+05]	79
3.1	Système robotique mobile d'expérimentation	84
3.2	Base mobile Pioneer 3DX	84
3.3	Scanner laser Hokuyo URG-04LX	85
3.4	Camera Kinect [©]	86
3.5	Relais USB-RELAY 2	86
3.6	Wattmètre Yocto-Watt	87
3.7	Schéma d'un module (figure tirée de [Pas10])	88
3.8	Structure d'un schéma ContrACT	89
3.9	Schéma de décomposition logicielle produite par ContrACT (figure tirée de [Pas10])	91
3.10	Phase d'apprentissage de la méthode de Chromo-Localisation LZA	93
3.11	Illustration des signatures laser issues de l'apprentissage et de la mesure . .	93
3.12	Illustration de la fonction de suivi de chemin de la méthode SMZ [LZ12] .	94
3.13	Illustration de la fonction d'évitement d'obstacle de la méthode SMZ [LZ12]	94
3.14	Illustration de la détection de murs et génération de chemin de la méthode COC	95
3.15	Diagramme UML de la carte d'environnement	98
3.16	Diagramme UML de la mission à réaliser : point de vue utilisateur	98
3.17	Diagramme UML du scénario de mission détaillé	99

3.18	Description de l'environnement de la mission	101
3.19	Vannes à inspecter	102
4.1	Diagramme causes-effets pour l'axe sécurité appliqué au système d'étude .	106
4.2	Diagramme d'interdépendance de l'axe sécurité appliqué au système d'étude	108
4.3	Evitement d'obstacle : Distance de sécurité, de perception et de réaction .	110
4.4	Illustration de la distance parcourue lors de la phase de perception	111
4.5	Illustration de distance nécessaire pour contourner un obstacle	112
4.6	Vitesses permettant d'assurer l'évitement d'obstacles et l'innocuité en fonction des capteurs	113
4.7	Énergie cinétique versus vitesse pour un objet en mouvement de 25 kg . .	115
4.8	Diagramme causes-effet pour l'axe énergie appliqué au système d'étude . .	116
4.9	Diagramme d'interdépendance de l'axe énergie appliqué au système d'étude	118
4.10	Architecture matérielle du système robotique étudié	120
4.11	Puissance de locomotion pour différentes vitesses du robot	125
4.12	Énergie de locomotion pour différentes vitesses et distances parcourues . .	127
4.13	Principe de discrétisation et de choix de la vitesse	128
4.14	Consommation énergétique des sonars pour différentes fréquences	129
4.15	Estimation de l'énergie consommée sur la batterie du robot (mission Avance Simple) : (a) Erreur de prédiction, (b) Résultats expérimentaux et théoriques	136
4.16	Estimation de l'énergie consommée sur la batterie du robot (mission Avance Simple) : (a) Erreur de prédiction, (b) résultats expérimentaux et théorique	137
4.17	Diagramme causes-effets pour l'axe durée appliqué au système d'étude . . .	139
4.18	Diagramme d'interdépendance pour l'axe durée appliqué au système d'étude	139
4.19	Diagramme causes-effets pour l'axe localisation appliqué au système d'étude	143
4.20	Diagramme causes-effets pour l'axe stabilité appliqué au système d'étude .	144
5.1	Rappel de la phase en ligne (P3) de l'approche de gestion des ressources guidée par la performance	162
5.2	Interface du simulateur <i>Matlab</i> de mission avec gestion de la performance	164

5.3	Illustration de la prise en compte d'un évitement d'obstacle EO dans le simulateur	165
5.4	Déroulement de la mission simulée	166
5.5	Évolution des marges de performance continues de la mission simulée . . .	167
5.6	Évolution de la vitesse du robot en fonction du temps et de la coordonnée linéaire x_l de position du robot	168
5.7	Evolution de la vitesse linéaire du robot au cours de la mission de patrouille	174
5.8	Puissance et énergie consommées sur les batteries du robot et du PC . . .	175
5.9	Trajectoire du robot lors de la mission de patrouille : elle est construite à partir des données odométriques recalées lors du passage à proximité des QR-codes	175
5.10	Évolution des marges de performances continues lors de la mission de patrouille	176
5.11	Première partie de l'algorithme de l'algorithme LZA (figure tirée de [Mou14])	193
5.12	Deuxième partie de l'algorithme de l'algorithme LZA (figure tirée de [Mou14])	194

Liste des tableaux

1.1	Les niveaux d'autonomie de décision pour un système artificiel (SA) proposés dans [PSW00]	11
1.2	Les niveaux d'automatisation de décision proposés dans [BFR14]	12
2.1	Tableau récapitulatif des nombres de tâches et OIT.	65
2.2	Les valeurs du vecteur I pour l'exemple de la figure 2.9	74
3.1	Liste des OIT par tâche	100
3.2	Description de la séquence d'objectifs de la mission cible	102
4.1	Classification des stratégies de sécurité [IIN03]	108
4.2	Puissances consommées par les composants statiques	124
4.3	Puissance instantanée estimée pour différentes AI à $v_r = 0.5$ m/s	130
4.4	Puissance instantanée du processeur pour différentes configurations de AI avec Avance Simple	133
4.5	Consommation des périphériques externes de l'ordinateur embarqué	134
4.6	Puissance instantanée globale consommée sur la batterie du PC pour différentes configurations de AI avec une Avance Simple	134
4.7	Tableau récapitulatif des extremums des erreurs d'estimation de la consommation énergétique sur la batterie du robot, la moyenne et l'écart type	137
4.8	Tableau récapitulatif des extremums des erreurs d'estimation de la consommation énergétique sur la batterie du PC, la moyenne et l'écart type	138
4.9	Tableau de composition des différentes AI et leurs références correspondantes	146

4.10	Code de couleur des vitesses maximales par zone	147
4.11	S��nario de Mission D��taill�� sans contraintes	148
4.12	SMD avec prise en compte de l'axe s��curit��	150
4.13	S��nario de Mission D��taill�� avec prise en compte des axes de performance s��curit�� et localisation	152
4.14	R��capitulatif du SMD sans prise en compte des axes de performance boo- l��ens et prise en compte de l'axe dur��e	154
4.15	R��capitulatif du SMD avec prise en compte de l'axe s��curit��	156
4.16	R��capitulatif du SMD avec prise en compte de tous les axes de performance bool��ens	157
4.17	D��tails de la SAR_0 calcul��e hors ligne	158
5.1	��v��nements g��n��r��s dans la mission simul��e	166
5.2	SAR_1 calcul��e lors de la mission simul��e	169
5.3	SAR_2 calcul��e lors de la mission simul��e	170
5.4	SAR_3 calcul��e lors de la mission simul��e	171
5.5	SAR_4 calcul��e lors de la mission simul��e	171
5.6	Tableau r��capitulatif des diff��rentes SAR calcul��es pour la mission en si- mulation	172
5.7	SAR_1	177
5.8	SAR_2	177
5.9	SAR_3	178
5.10	Tableau r��capitulatif des diff��rentes SAR calcul��es au cours de l'exp��ri- mentation	178
5.11	Tableau r��capitulatif des consommations pour les diff��rents OIT (Table 3.1)	195
5.12	$RSMD_1$ pour la mission simul��e	198
5.13	$RSMD_2$ pour la mission simul��e	199
5.14	$RSMD_3$ pour la mission simul��e	200
5.15	$RSMD_4$ pour la mission simul��e	201

5.16 $RSMD_1$ pour la mission réelle	202
5.17 $RSMD_2$ pour la mission réelle	203
5.18 $RSMD_3$ pour la mission réelle	203

Glossaire des abréviations

A	Activité
AI	Alternative d'Implémentation
c	contrainte
dis_A	distance parcourue associée à une activité
dur_A	durée d'une activité
I_A	vecteur d'information d'une activité
M	Marge de performance
n_{act}	nombre d'activités
NGA	Nombre Global de combinaisons possibles d'Alternatives d'implémentation
n_{OIT}	nombre d'OIT
n_T	nombre de Tâches
O	Objectif
OIT	Option d'Implémentation de Tâche
$Perf_{SR}$	Performance du Système Robotique
$Perf_C$	Performance Contrainte (objectif)
RSMD	Reste du Scénario de Mission Détaillé
SAR	Solution d'Allocation de Ressources
SMD	Scénario de Mission Détaillé
SR	Système Robotique
T	Tâche
US	UltraSon
x_A	coordonnée linéaire du début d'une activité

Glossaire des principaux concepts

Autonomie	Capacité individuelle ou collective de décider et d'agir de manière cohérente sans contrôle ou intervention extérieure.
Autonomie décisionnelle	Capacité à choisir les actions à réaliser pour atteindre les objectifs fixés.
Autonomie comportementale	Capacité à réaliser l'action choisie en fonction des ressources matérielles et logicielles disponibles et opérationnelles à un instant donné.
Décision multicritère	Action par laquelle un choix, satisfaisant un ensemble de critères prédéfinis, est réalisé parmi un ensemble d'alternatives possibles.
Performance	Grandeur subjective ou objective, pouvant être multidimensionnelle, permettant d'estimer/évaluer le résultat d'une action au regard des moyens mis en œuvre et de(s) l'objectif(s) fixé(s).
Inducteur de performance	Levier sur lequel on intervient pour influencer la performance d'un système. Les inducteurs de performance permettent de piloter la performance d'un système.
Indicateur de performance	Donnée quantifiée qui mesure l'efficacité de tout ou partie du système robotique.

Introduction

Si le 20^{ème} siècle a vu l'avènement du monde numérique, en s'inspirant de la formule attribuée à A. Malraux, nous pouvons sans doute prophétiser que le 21^{ème} siècle sera celui des Systèmes Cyber-Physiques (SCP).

Les Systèmes Cyber-Physiques se déclinent dans des domaines aussi variés que l'aéronautique (Air Traffic Management) [Poo+09], la santé [Lee+12a], l'énergie (Smart Grid) [Kar11], les systèmes de transports intelligents (Cyber-Physical Vehicle System) [BA15] ou la robotique [Chi+13]. Ils associent des ressources logicielles de calcul embarqué et de communication (côté Cyber), à des ressources matérielles de perception, de contrôle, d'actionnement et de stockage d'énergie (côté Physique) [BA15].

Cependant les SCP se distinguent des systèmes cybernétiques classiques du 20^{ème} siècle par leur complexité, leur intégration dans un environnement dynamique, et par le fait que la qualité et la continuité du service visé exclut d'adopter une approche de conception réductrice juxtaposant les champs de connaissances habituels. En effet, l'atteinte des objectifs de "stabilité, performance, fiabilité, robustesse et efficacité" [KK12] escomptés impose de s'orienter vers une démarche de conception plus holistique que la démarche incrémentale usuelle qui serait trop coûteuse ou tout simplement vouée à l'échec [Lee08].

Les SCP se caractérisent aussi évidemment par une autonomie décisionnelle accrue. L'affirmation de la dimension matérielle les distingue du concept d'Autonomic Computing [Hor01] [Mur04] proposé par IBM au début des années 2000. Cela les rapproche, voir les confond, avec les Systèmes Autonomiques (SA) que vise à développer la NASA [Tru+09] pour l'exploration spatiale. Un système autonome classique se caractérise "simplement" par sa capacité à développer un certain niveau d'auto-gouvernance lui permettant de réaliser avec succès les tâches qui lui sont dévolues. Un SA quant-à-lui y ajoute une dimension d'auto-management qui doit permettre d'assurer que le système reste opérationnel et performant. Cette distinction peut paraître ténue mais elle est cependant fondamentale lorsque l'on s'intéresse, comme dans le cadre de cette thèse, à des systèmes devant assurer des tâches complexes, durant de longues périodes, sans aucune intervention humaine. Le déploiement de SCP/SA s'articule autour de différents objectifs [Tru+09] que nous

adresserons plus ou moins en profondeur dans le travail que nous allons présenter :

- L'Auto-Optimisation (*self-optimizing*) qui traduit la capacité qu'un système doit avoir à mesurer sa performance actuelle pour la comparer à la performance visée de façon à pouvoir mettre en place, si cela est possible, des politiques d'amélioration.
- L'Auto-Guérison (*self-healing*) qui traduit la capacité qu'un système a de détecter ou prédire des fautes de façon à pouvoir mettre en place des mécanismes de recouvrement adaptés.
- L'Auto-Protection (*self-protecting*) qui traduit la capacité qu'un système doit avoir à se défendre d'attaques accidentelles ou mal attentionnées.
- L'Auto-Configuration (*self-configuring*) qui traduit la capacité qu'un système doit avoir pour s'adapter aux circonstances (évolution du système, de son environnement, de sa mission) tout en assurant la réalisation des objectifs précédents.

Ces objectifs se traduisent au travers d'un ensemble de propriétés, parfois recouvrantes, que les SCP/SA se devraient idéalement de vérifier [Gun+14]. Parmi celles-ci, les travaux présentés dans ce manuscrit relèvent des propriétés suivantes :

- Sûreté de fonctionnement (*dependability*) : Capacité d'un système à fonctionner durant son opération sans dégradation notable de sa performance. Elle permet aux utilisateurs d'avoir une confiance justifiée dans le service délivré.
- Fiabilité (*reliability*) : capacité d'un système à fonctionner correctement en présence de conditions défavorables, pendant une durée déterminée et dans une plage spécifiée.
- Robustesse (*robustness*) : capacité d'un système à garder une configuration stable même en présence de défaillances internes¹.
- Sécurité (*safety*) : Capacité d'un système à ne pas engendrer de danger envers lui-même ou son environnement.
- Prédicibilité (*predictability*) : Capacité d'un système à garantir son comportement futur sur une plage temporelle importante.
- Adaptabilité (*adaptability*) : capacité d'un système à changer d'état en ajustant sa propre configuration de façon à répondre aux variations de son environnement.
- Reconfiguration (*reconfigurability*) : Capacité d'un système à changer ses configurations en cas de défaillance ou sur demande, interne ou externe.

1. La question de la robustesse pour l'automatique se traduit par la capacité à garantir la convergence du système asservi, malgré des erreurs d'estimation des paramètres du modèle. Par extension, à un niveau d'abstraction plus élevé (mission), les défauts paramétriques des modèles s'incluent sous le terme de "défaillance interne".

- Efficacité (*efficiency*) : capacité d'un système à utiliser au mieux ses ressources (logicielles, matérielles, énergétiques, temporelle, etc.) de façon à délivrer le service demandé.
- Résilience (*resilience*) : capacité d'un système à maintenir une qualité de service acceptable même en présence de défaillances internes ou externes.
- Durabilité (*sustainability*) : capacité d'un système à fonctionner durablement tout en maintenant sa qualité de service, en renouvelant ses ressources et en les utilisant efficacement.

L'atteinte de ces objectifs, et la vérification de ces propriétés, impliquent que le système connaisse son état interne (*self-aware*) ainsi que les ressources disponibles et le statut qui leur est associé (opérationnelle, défaillante, utilisable, inopérante, etc.). Il doit aussi être capable d'identifier les conditions environnementales courantes (*environment-aware*) [Tru+09].

Parmi les SCP/SA, ce travail se focalise plus particulièrement vers les missions autonomes complexes (recouvrant un ensemble de tâches), de longue durée, en robotique mobile, au sein d'un environnement connu mais dynamique. Certains travaux parlent alors de Robotique Autonome (*Autonomic Robotic*) [Ins12] [Ins13]. Sous bien des aspects, l'étude présentée pourrait aussi s'appliquer aux systèmes de transport intelligents.

Il est de bon sens d'énoncer qu'un système robotique mobile se doit "d'assurer sa fonction de locomotion tout en garantissant l'atteinte des objectifs de la mission qui lui est dévolue tout en satisfaisant des contraintes sécuritaires et d'efficacité" [BA15]. Il est par ailleurs évident que le système doit être capable de "développer un certain niveau de robustesse vis-à-vis de perturbations externes (modification de l'environnement), des bruits capteurs ou des erreurs de modélisation", tout en restant capable de fonctionner sous un large spectre de conditions expérimentales" [BA15].

Pour atteindre ces objectifs, en s'appuyant sur l'analyse de Bradley et Atkins [BA15] un ensemble de défis doivent être adressés :

- Interaction avec l'Homme : un système robotique est évidemment développé pour apporter des réponses à des besoins humains. Même pour les missions fortement autonomes qui nous intéressent, l'utilisateur (*end-user*) est le premier prescripteur de la mission. C'est donc à lui de la définir et d'en préciser les performances attendues. En retour il paraît aussi important que le système robotisé soit à même de juger de la faisabilité de la mission définie.
- Tolérance aux fautes : c'est un élément indispensable de la sûreté de fonctionnement qui doit être déployé lors de missions de longue durée où le système robotique devra potentiellement faire face à des dérives comportementales et des défaillances

matérielles et/ou logicielles. La tolérance aux fautes nécessite la mise en place de mécanismes de détection et d'isolation de la faute et de recouvrement (*Fault Detection Isolation and Recovery*).

- Management de l'énergie : c'est évidemment l'un des défis majeurs de la robotique mobile lorsque l'on considère des missions de longue durée. Le système doit disposer d'assez d'énergie pour mener à bien sa mission.
- Management des ressources matérielles et logicielles : le système robotique dispose d'un ensemble de ressources matérielles (capteurs, actionneurs) et logicielles (algorithmes de contrôle, d'analyse d'image, etc.) assurant les fonctionnalités robotiques nécessaires à la mission, et dont les domaines de fonctionnement sont connus. Il se doit de les utiliser au mieux en fonction des objectifs de la mission, de l'environnement traversé et évidemment de leur coût énergétique.
- Scénarisation de la mission : la vérification de la faisabilité de la mission nécessite d'avoir la capacité, tout au long de son exécution, de projeter le système robotique dans l'avenir. Pour cela la notion de scénario nous semble indispensable. Il est cependant évident que tous les scénarii d'une mission ne peuvent être envisageables. En lien avec la tolérance aux fautes, il faut donc doter le robot mobile de mécanismes lui permettant de tolérer certaines déviations au scénario planifié. Dans le cadre de ce travail, nous n'aurons pas la capacité de changer de scénario, mais tant que celui-ci restera réalisable avec les ressources disponibles, nous serons capables de les choisir pour que la mission soit menée à son terme.

Force est de constater que peu de travaux en robotique mobile s'intéressent actuellement aux missions autonomes longues et multi-tâches. La gestion de l'énergie reste un problème majeur peu abordé. L'autonomie véritable sans supervision humaine, où le robot est livré à son libre arbitre durant de longues périodes reste rare. L'analyse des missions robotiques de sauvetage et militaires menées par Carlson [CM05a] a démontré la fiabilité limitée des systèmes robotisés. Les missions robotiques phares de la NASA sur la planète Mars restaient très fortement supervisées par le centre de contrôle terrestre et l'autonomie de navigation, si elle a bien été déployée, était exécutée à très faible vitesse. Même si l'opérateur reste une source d'erreur non négligeable, en raison de ses capacités de raisonnement, d'abstraction et d'anticipation, il reste à ce jour le plus à même pour gérer les déviations de scénario. On peut cependant s'interroger sur son aptitude à évaluer la faisabilité d'une mission où se demander sur quels critères objectifs il choisi à un instant donné d'utiliser une configuration logicielle et/ou matérielle, si ce n'est le poids de l'expérience.

Le contexte de cette étude ayant été peu à peu affiné, il est maintenant possible d'énoncer la problématique centrale de ce manuscrit :

Quelles ressources matérielles et logicielles un système robotique mobile autonome doit-il mobiliser et paramétrer tout au long de sa mission de façon à garantir que l'exécution de cette dernière satisfasse un ensemble de contraintes de performance ? Le mécanisme d'allocation développé devra être à même de maintenir ces objectifs (faisabilité mission, contraintes de performance) lors de l'exécution de la mission malgré l'apparition de dérives comportementales et/ou en présence d'événements contraires tels que des défaillances matérielles ou logicielles.

La réponse que nous apportons s'organise de la façon suivante au sein de ce manuscrit.

Le premier chapitre revient sur le concept d'autonomie qui inévitablement conduit à la notion de choix. Choisir nécessite de disposer d'un critère objectif et c'est la performance que nous avons retenue. Bien qu'elle soit couramment employée en robotique ce concept reste mal défini. Les travaux menés en productique nous permettront de mieux la cerner. L'analyse de son utilisation en robotique industrielle et en robotique mobile démontrera que cette dernière ne dispose pas d'un cadre d'étude largement accepté. Enfin, la prise de décision relevant le plus souvent de plusieurs critères à la fois, la dernière partie balaie les principales techniques d'allocation de ressources multicritères développées.

Le second chapitre constitue le cœur de ce document. Dans un premier temps un cadre d'étude de la performance est proposé pour la robotique mobile. Nous y distinguerons les axes principaux, dénominateurs communs à toute mission robotique mobile, des axes secondaires centrés sur les besoins utilisateurs. Si les premiers recouvrent la **Sécurité**, la **Stabilité**, la **Localisation** et l'**Énergie**, nous retiendrons pour les seconds la **Durée** de la Mission. Avant d'aborder la description de la méthodologie d'allocation des ressources développée, une classification des performances est proposée et le concept de marge de performance est introduit. La démarche d'allocation est ensuite détaillée. Elle se décompose en trois phases. Tout d'abord une phase préliminaire hors ligne, statique où, pour chaque axe d'intérêt les inducteurs de performance sont identifiés, les modèles d'estimation construits, et les contraintes de performance projetées sur le scénario de la mission. La deuxième phase hors ligne concerne la conclusion sur la faisabilité de la mission vis à vis des différentes contraintes de performance et de l'allocation des ressources matérielles et logicielles aux différentes tâches robotiques. Ce problème s'apparentant à un problème de type sac-à-dos multicritères, nous montrerons alors comment trouver efficacement et rapidement une bonne solution au problème d'allocation sous contraintes qui nous intéresse. La même approche sera utilisée durant la troisième phase dynamique en ligne, lors de l'exécution de la mission pour trouver à nouveau une solution d'allocation en temps

réel en cas de dérive de performance ou de défaillance matérielle ou logicielle.

Le troisième chapitre permet de présenter le cadre expérimental qui nous permettra d'évaluer la méthodologie proposée. Le système robotique et son équipement matériel sont tout d'abord décrits. L'architecture de contrôle ensuite ainsi que les algorithmes utilisés pour déployer les fonctionnalités robotiques. Dans un dernier temps, une mission de patrouille au sein des couloirs du laboratoire est retenue. Les caractéristiques de cet environnement sont définies.

Le quatrième chapitre détaille la mise en œuvre statique de la méthodologie d'allocation des ressources matérielles et logicielles proposée sur la mission de patrouille présentée. Cette dernière doit satisfaire un ensemble de contraintes de performance. Dans le cadre de cette étude nous nous focaliserons principalement sur les axes Sécurité, Énergie et Durée. La Sécurité se déclinera au travers d'indicateurs d'évitement d'obstacle et d'innocuité. L'énergie constituera une partie importante de l'étude présentée. Une modélisation générique du modèle de consommation est proposée pour la batterie du robot et pour celle du laptop. Par ailleurs un modèle énergétique de locomotion est aussi établi et évalué. C'est l'impact énergétique des ressources matérielles et logicielles qui guidera principalement le mécanisme d'allocation des ressources. Enfin l'axe Durée nous permettra d'assurer le déplacement du robot. Ainsi un scénario détaillé de la mission est construit où des ressources matérielles et logicielles précises sont affectées aux différentes activités de la mission si celle-ci est réalisable.

Le cinquième et dernier chapitre se focalise sur la phase dynamique de la méthodologie qui est conduite durant l'exécution de la mission robotique. Dans un premier temps la simulation permet de mieux comprendre comment les marges de performance proposées permettent de robustifier notre démarche en la rendant moins sensible aux dérives de performance, mais aussi comment nous pouvons, en présence de défaillances matérielles ou logicielles, retrouver en temps réel une solution d'allocation des ressources vérifiant les contraintes de performance définies, si elle existe. Dans un second temps, des relevés expérimentaux viendront corroborer l'analyse faite en simulation.

Enfin pour clore ce document, après avoir rappelé les principales caractéristiques et originalités du travail présenté nous évoquerons évidemment les limitations de celui-ci ainsi que quelques pistes qui pourraient permettre de les repousser. Nous montrerons alors que la méthodologie présentée peut, à notre avis, avoir de multiples applications en robotique mobile autonome.

Chapitre 1

De l'autonomie à la décision multicritère : État de l'art

Sommaire

1.1	Autonomie	8
1.1.1	Modes/niveaux d'autonomie	10
1.1.2	Autonomie comportementale	12
1.2	Performance	16
1.2.1	Concept de performance	16
1.2.2	Performance en robotique	26
1.2.3	Conclusion	36
1.3	Décision multicritère	37
1.3.1	Mission et décision multicritère	38
1.3.2	Techniques de résolution	41
1.4	Positionnement	46

Le concept d'autonomie relève nécessairement la notion de capacité de décision (choix). Cela apparaît clairement dans [LC07] où les auteurs considèrent que le plus haut degré d'autonomie est assuré si ce dernier est au sommet de tous les nœuds (acquisition, analyse d'information, **décision/sélection** et mise en œuvre) formant un système. La décision, quant à elle, implique le choix du "quoi faire" (autonomie décisionnelle) et du "comment faire" (autonomie comportementale). Ces choix sont toujours motivés par un ou souvent plusieurs critères objectifs caractérisant la performance attendue du système autonome. Brainov *et al.* ramènent dans [BH01] l'autonomie à la stabilité des performances d'un agent. Dans la même logique de raisonnement, Hexmoor affirme dans [Hex00] que l'autonomie n'a de sens que si l'agent est capable de réaliser la tâche qui lui est allouée. Cette réflexion est confirmée dans [Cho07] où Chopinaud met en évidence les relations entre

l'autonomie, la prise de décision et la performance (Figure 1.1). Nous pouvons illustrer les liens entre ces différents concepts dans la figure suivante.

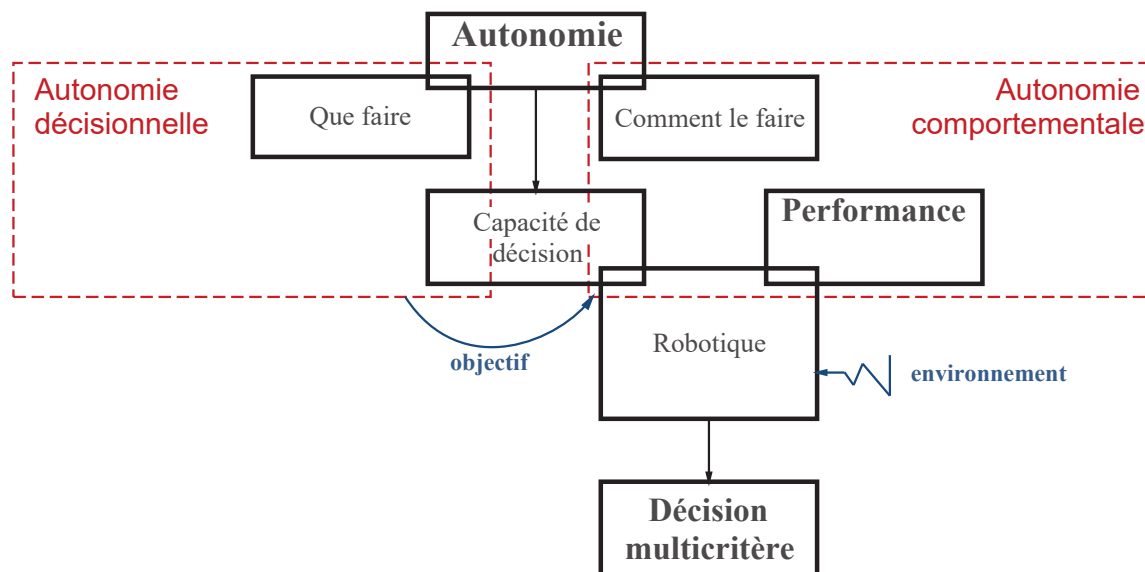


FIGURE 1.1 – Liens entre les concepts d'autonomie, de décision, et de performance

Autonomie, performance et décision, sont donc autant de concepts fortement liés dans le cadre de la robotique mobile qui nous intéresse. Ce premier chapitre va nous permettre de mieux appréhender ces notions centrales pour notre étude. De par leur large périmètre, outre l'automatique et la robotique, nous nous rapprocherons d'autres champs disciplinaires, tels que la productique, la biologie ou la philosophie, pour mieux les cerner.

1.1 Autonomie

L'autonomie est un concept philosophique très difficile à définir. Il a été projeté sur plusieurs domaines y compris celui de la robotique mobile. Certaines définitions peuvent s'avérer absurdes et difficilement applicables sans une prise en compte du domaine d'application. Nous allons dans cette partie balayer les définitions les plus pertinentes et mettre en évidence les liens entre ces principaux concepts.

Tout d'abord, la notion d'autonomie est un concept existant depuis l'antiquité. Il a été identifié dans l'étymologie grecque par les termes *autos* (soi) et *nomos* (règle ou loi). Il a été appliqué dans des cités grecques où les citoyens créaient leurs propres lois tout en s'opposant à l'autorité du gouvernement [Ste95]. Ce terme a souvent été associé ou confondu avec celui d'indépendance qui signifie selon Steel "*être indépendant de*". L'origine de ce terme relève alors de la définition de la notion de capacité de décision face à des contraintes extérieures. A contrario, en biologie, les systèmes de structures

biologiques sont considérés comme autonomes puisqu'ils ne sont pas construits par un organisme extérieur [Ste95; DI08] ce qui bien évidemment les distingue des systèmes artificiels où ces derniers ne peuvent être construits par eux mêmes. En dehors de cet aspect sociologique et biologique qui relèvent du vivant, l'autonomie a été projetée sur différents domaines pour les systèmes artificiels.

Plusieurs nuances de l'autonomie sont apparues comme l'automatisation et l'autonomique. Dans ses échanges avec Steel [Ste95] en 1992, Tim Smither écrit : *"Il est utile de comparer l'autonomie avec le concept des systèmes automatiques. La signification d'automatique provient de l'étymologie du terme cybernétique, qui vient du grec pour l'auto-direction. En d'autres termes, les systèmes automatiques sont auto-régulés, mais ils ne créent pas les lois qui permettent cette régulation. Celles-ci sont données à eux, ou construites en eux. Ils se dirigent le long d'un chemin donné, se corrigent et se compensent sous l'effet de perturbations externes. D'autre part, les systèmes autonomes sont ceux qui développent, par eux-mêmes, les lois et les stratégies selon lesquelles ils règlent leur comportement : ils sont auto-dirigés et auto-régulés. Ils déterminent les chemins qu'ils suivent et comment les suivre"*. Cette description révèle le contraste entre *autonomie* et *automatisme* en montrant la différence entre la génération de lois et leur suivi. La notion de *régulation de comportement*, exposée dans ce communiqué, est aussi évoquée dans [Cas95].

Truskowski *et al.* montrent dans [Tru+09] la différence entre l'autonomie et l'automatisation bien qu'ils affirment que plusieurs définitions existent. L'autonomie signifie pour eux l'auto-gouvernance et l'auto-direction. Le système choisi alors lui même ses buts et prend les décisions menant au succès des tâches à exécuter. En revanche, l'automatisation signifie l'auto-management par la prise de décisions aboutissant au bon fonctionnement du système. Ils excluent donc tout choix autonome des objectifs à atteindre. L'autonomie nécessite donc le recours à l'automatisation. Ils rajoutent aussi que le concept d'autonomique découlant du terme autonomie, désignent la capacité à s'auto-maintenir, s'auto-configurer, s'auto-optimiser et s'auto-protéger [Hor01] [Com03].

Il faut noter que dans ces définitions, les notions d'objectif et de performance ne sont pas présentes. Même la notion de but évoquée dans [Tru+09] est considérée comme le résultat d'un choix et non plus comme un critère de choix. Ceci est un manque considérable si l'on veut projeter ces définitions sur le domaine de la robotique où l'objectif est défini par l'extérieur. Ces définitions sont alors nécessaires mais elles ne sont pas suffisantes pour définir l'autonomie dans le cadre de notre étude.

En robotique, l'autonomie a été largement étudiée tout en restant assez ambiguë et sans définition communément acceptée [Dal03]. Cependant la définition proposée par [BH03a] peut être appliquée quel que soit le domaine d'utilisation visé : *"Le concept d'autonomie*

est apparenté à une capacité individuelle ou collective de décider et d'agir de manière cohérente sans contrôle ou intervention extérieure". Bien que cette définition met en évidence la notion de décision, elle ne considère pas la motivation indispensable pour guider la décision. Cela a été souligné dans la définition de Braynov et Hexmoor [BH03b] de l'autonomie décisionnelle en indiquant qu'il est essentiel de savoir ce que l'on veut faire (but) et quels sont les choix possibles. Ils rajoutent aussi que la mesure du succès de l'agent (atteinte du but) est un des constituants en relation avec l'autonomie.

Les différentes définitions évoquées dans cette partie démontrent que le concept de l'autonomie relève principalement de la capacité de décision en étant indépendant de l'extérieur et ceci sous des perturbations et contraintes extérieures. Cependant, le critère de décision n'est pas toujours explicitement souligné. Ce critère est bien évidemment le service attendu du robot exprimé en terme(s) d'objectif(s) ou but(s). Nous pouvons alors définir l'autonomie dans le contexte de notre étude comme la capacité de décision et de prise de décision sans intervention extérieure dans le but d'effectuer une mission avec des contraintes de performance prédéfinie tout en s'adaptant aux différents aléas internes ou externes.

L'autonomie en robotique se décline sous deux facettes : l'autonomie décisionnelle qui consiste à choisir l'action à faire et l'autonomie comportementale (de niveau plus bas) qui détermine la façon de réaliser cette action. La prise en compte de ces deux facettes a amené à définir plusieurs niveaux d'autonomie en partant du plus élémentaire où le robot n'est qu'un simple système d'exécution des ordres de l'opérateur qui décide l'action à réaliser et la façon de les réaliser, jusqu'à l'autonomie totale où le robot a la capacité de choisir ses propres objectifs et les moyens utilisés pour les atteindre. Une analyse plus approfondie de l'autonomie comportementale qui nous intéresse dans ce manuscrit sera réalisée dans le prochain paragraphe après avoir balayé les différents niveaux d'autonomie.

1.1.1 Modes/niveaux d'autonomie

Le niveau d'implication de l'Homme est primordial pour la définition de l'autonomie. Ceci se manifeste clairement dans la définition proposée par Parasuraman de l'autonomie dans [PSW00], qui énonce qu'un système autonome accomplit (partiellement ou totalement) une fonction qui était auparavant, ou éventuellement pourrait être réalisée (partiellement ou totalement), par un opérateur humain [PR97]. Ainsi, une échelle de 10 niveaux d'autonomie est proposée dans le tableau 1.1.

L'implication de l'Homme dans la décision du système est inexistante pour le niveau d'autonomie le plus haut (10). Tout comme dans [Goo08], un système est donc considéré comme autonome s'il est capable de réaliser des tâches sans interaction avec l'exté-

rieur. Cette interaction inclue tout échange d’informations et de ressources avec l’extérieur (contrôle de l’humain, énergie, données de perception, etc.). Ainsi, le degré d’autonomie décisionnelle décroît jusqu’au niveau le plus bas où le système n’apporte aucune assistance à l’homme, qui le contrôle complètement. Cette définition ne peut être pratiquement valide puisque le robot ne peut décider de l’objectif de son utilisation et qu’à un moment ou un autre on lui a communiqué ses objectifs et on l’a doté de certaines ressources lui permettant d’assurer les fonctionnalités nécessaires à leur réalisation.

Tableau 1.1 – Les niveaux d’autonomie de décision pour un système artificiel (SA) proposés dans [PSW00]

Basse	1	le Système Artificiel (SA) n’offre aucune assistance
	2	le SA présente un ensemble complet de possibilités
	3	le SA présente un ensemble restreint de possibilités
	4	le SA suggère une possibilité
	5	le SA exécute sa suggestion si l’humain approuve
	6	le SA accorde un délai pour un éventuel veto avant l’exécution
	7	le SA agit puis informe l’humain
	8	le SA informe l’humain si ce dernier le demande
	9	le SA informe l’humain s’il (le SA) le décide
Haute	10	le SA décide de tout, agit seul et ignore l’humain

Clough distingue clairement autonomie et automatisme dans [Clo02]. Il qualifie de système autonome un système qui a son libre arbitre et peut faire ses choix librement sans influence de l’extérieur. Pour lui, un système automatique ne dispose d’aucun degré de liberté et opère conformément au programme imposé. Le même classement de niveau d’autonomie dans le tableau 1.1 est trouvé dans [BFR14] en y rajoutant une taxonomie des niveaux d’automatisation proposée par [EK99]. Celle ci est organisée selon quatre fonctions génériques :

- Surveillance : suivi de l’état du système,
- Génération : formulation des stratégies pour atteindre les objectifs,
- Sélection : décider d’une option ou d’une stratégie,
- Implémentation : mise en œuvre de l’option choisie.

Cela a permis de proposer une taxonomie de 10 niveaux d’automatisation (table 1.2). Elle peut être applicable à un large éventail d’applications. Ces classements (autonomie de décision et automatisation) sont utilisés pour évaluer l’autonomie dans différents domaines y compris en robotique.

Un système robotique autonome dispose généralement de plusieurs ressources matérielles et logicielles. Ainsi, le plus haut degré d'autonomie implique la nécessité de doter le système robotique de la capacité de décider/choisir ses objectifs, la manière de les atteindre (stratégies) et enfin les mettre en œuvre (implémentation). Le travail présenté dans cette thèse se situe dans le cadre de l'autonomie du choix d'implémentation qui est appelée "autonomie comportementale" depuis les années 90.

Tableau 1.2 – Les niveaux d'automatisation de décision proposés dans [BFR14]

Niveaux d'automatisation	Description
Contrôle manuel	L'homme prend le contrôle, génère les options, sélectionne les options (décide) et met en œuvre les options choisies
Aide à l'action	L'automatisme aide l'humain exécutant les actions sélectionnées. L'humain peut réaliser quelques actions
Traitement par lot	L'homme génère et sélectionne les options ; celles ci sont réalisées automatiquement (ex. régulateur de vitesse en voiture)
Contrôle partagé	L'humain et l'automatisation génèrent des options de décision possibles. L'humain sélectionne les options. Leur mise en œuvre est une tâche partagée.
Aide à la décision	L'automatisation génère des options de décision. L'humain peut en choisir une qui sera exécutée par l'automatisation.
Prise de décision mixte	L'automatisation génère une option, la sélectionne, et l'exécute si l'humain l'autorise. L'humain peut approuver l'option choisie par l'automatisation, en sélectionner une autre, ou générer une autre option.
Système rigide	L'automatisation fournit un ensemble d'options et l'humain sélectionne l'une d'entre elles. Une fois sélectionnée, l'automatisation exécute la fonction.
Prise de décision automatisée	L'automatisation sélectionne et effectue une option. L'humain peut paramétrer les alternatives générées par l'automatisation.
Contrôle supervisé	L'automatisation génère des options, sélectionne et réalise une option souhaitée. L'homme observe le système et intervient si nécessaire.
Automatisation complète	Le système effectue toutes les actions.

1.1.2 Autonomie comportementale

Nous allons commencer par détailler le concept d'autonomie comportementale, ses relations avec l'extérieur ainsi que les éléments qui amorcent les changement de compor-

tement. Ceci mènera cette étude vers l'implication importante du concept de performance dans la notion d'autonomie.

1.1.2.1 Concept de l'autonomie comportementale

Dalgarrondo considère dans [Dal03] qu'un système disposant d'un ensemble de lois (déjà développées) est autonome s'il est capable de réagir à des événements externes en sélectionnant le comportement le plus adapté. Plusieurs niveaux d'adaptation sont alors considérés. Celui ci peut aller du simple ajustement de quelques paramètres jusqu'à l'apprentissage. Cependant, la sélection des lois à appliquer ne peut être seulement liée aux événements extérieurs mais aussi aux objectifs de la mission à réaliser.

L'autonomie et en particulier celle comportementale est évidemment l'une des caractéristiques essentielles des êtres vivants. Par conséquent la biologie a souvent été une source d'inspiration pour le développement des robots mobiles tant sur leur concept que sur leur comportement, en mono-robot ou en flottille. Cette capacité est considérée comme un critère essentiel des êtres vivants.

Steel et DiPaolo s'en sont inspiré dans [Ste95] et [DI08] pour définir des agents autonomes intelligents. L'intelligence est relative à la capacité d'adaptation comportementale de ces agents (auto-maintenance). L'autonomie est ainsi considérée pour eux comme la capacité d'une entité/système à se doter de lois et de stratégies lui permettant de contrôler (choisir) son comportement. Ce choix est constitué des sous-buts et/ou de moyens à mettre en œuvre pour d'atteindre le but fixé.

De plus, Calo *et al.* considèrent dans [CFK16] que contrairement à la télé-programmation ou à la pré-programmation, la "vraie" autonomie (*true autonomy*) implique nécessairement d'être capable de générer ses propres heuristiques (Apprentissage) pour prendre une décision pertinente. Froeze énonce dans [FVI07] que la différence de traitement de l'information et de raisonnement entre un être vivant (humain en particulier) et un système artificiel réside dans le fait que le premier base ses décisions sur le comportement à adopter sur des heuristiques ou des analogies alors que le deuxième s'appuie sur des règles de calcul prédéfinies. L'autonomie comportementale est alors définie comme la capacité à réaliser des tâches sans aide sous des contraintes et face à des événements prédéfinis. Toutefois, cette définition qui peut s'appliquer à des systèmes artificiels reste très restrictive pour un être humain. En effet, pour ce dernier l'autonomie n'est pas liée à la connaissance des événements auxquels il aura à réagir.

Le concept d'autonomie reste cependant relatif. Selon Di Paolo dans [DI08], un système dont l'identité est entièrement conçue par un extérieur et qui ne peut par lui même générer

ses constituants, ne peut que suivre les lois définies durant sa conception, quelles que soient son adaptativité et sa flexibilité. Pour qu'un système puisse générer ses propres lois, il doit être en mesure de se construire lui-même, à un certain niveau. La question qui se pose alors logiquement est de définir le niveau de granularité à partir duquel le système peut être considéré comme entièrement autonome ? D'où la relativité du concept d'autonomie.

Di Paolo poursuit cette réflexion en considérant que l'élaboration d'un système artificiel doté de la capacité d'auto-construction reste illusoire puisqu'il faut bien qu'à un moment donné on l'ait doté d'un minimum de capacités et de ressources. Cette attitude implique la prise en compte de toutes les conditions nécessaires pour l'auto-constitution. Ainsi un minimum de ressources et de capacités doivent être établies par l'extérieur. Cette rationalité dans le traitement du sujet de l'autonomie est affirmée dans [Sim96] par Simon qui énonce que les actions d'un programme ou d'un robot sont limitées par les informations dont il dispose, par les temps de calcul possible et par les limitations de ses algorithmes.

Si l'on se projette dans le cadre de ce manuscrit qui s'intéresse à la robotique mobile, les capacités et ressources matérielles (actionneurs, contrôleurs, capteurs, unité de calcul, etc.) et logicielles (algorithmes, drivers, etc.) ne peuvent être créés ou générés par le robot lui-même. Par contre leur exploitation, agencement et organisation peut être faite d'une façon autonome.

Nous pouvons alors définir l'autonomie comportementale, dans notre contexte, par la capacité de réaction et d'adaptation des ressources matérielles et logicielles dont le robot dispose, afin de réagir à des événements internes et externes et ceci dans le but d'atteindre une finalité donnée.

L'analyse précédente sur l'autonomie en général et l'autonomie comportementale en particulier, a réaffirmé que la prise de décision autonome est inévitablement conditionnée par la performance. Cette dernière représente le critère permettant de faire les choix et de prendre des décisions.

En robotique mobile, les missions complexes sont le plus souvent supervisées par un opérateur. L'Homme, de part ses capacités cognitives d'anticipation, d'adaptation et de synthèse lui permettent de guider le robot pour réaliser des tâches élaborées, dans un environnement dynamique, face à des événements imprévus. La performance maximale est atteinte quand l'homme est fortement impliqué (exemple de la téléopération où la charge décisionnelle est principalement rapportée à l'opérateur) [Sel+06a]. Cependant, d'une part ce lien de téléopération (sous-marin, exploration spatiale, etc.) est contraignant et fonctionnellement limitatif. Par ailleurs, rien ne garantit objectivement que les choix réalisés par l'opérateur ont été les plus adaptés à la mission fixée aux regards de contraintes de

performance imposées. Nous allons approfondir dans la suite cette analyse afin de montrer l'importance du concept de la performance dans le domaine de la robotique autonome.

1.1.2.2 De l'autonomie comportementale à la performance

Il est évident que tout système robotique et en particulier celui autonome et mobile est conçu pour répondre à un besoin spécifique. L'impact de ce besoin apparaît implicitement ou explicitement dans plusieurs définitions et descriptions de l'autonomie. Ce service est la raison d'être du système robotique. Steel dans [Ste95] définit un agent, comme étant un système en relation avec d'autres systèmes, ou l'environnement, assurant une fonction envers un autre agent ou système. De plus, il doit être capable de s'auto-maintenir pour assurer sa viabilité.

Dans [Dal03], Dalgarrondo considère l'autonomie en robotique comme un concept complexe et non absolu qui est lié, entre autres, aux tâches qu'il doit effectuer. L'autonomie permet au robot de répondre aux événements et perturbations externes en exploitant ses ressources internes ainsi que ses fonctionnalités, ses capacités et son énergie. Cependant, les événements et perturbations internes ne sont pas prises en compte dans la définition proposée. Cette réponse va nécessairement impacter la performance du robot en modifiant les ressources et fonctions utilisées. Il montre aussi que l'environnement ajoute une complexité à la mise en œuvre du concept de l'autonomie. Il est à la fois une des principales sources de contraintes et d'informations, imposant au système robotique de s'adapter.

Cette capacité d'adaptation constitue la caractéristique principale de l'autonomie d'un robot. Dalgarrondo considère qu'elle permet de répondre et "*se mettre en harmonie*" avec les conditions de vie de tout être vivant. Elle permet alors d'assurer un certain niveau de performance dans la réalisation de l'objectif, malgré les perturbations externes et en fonction des ressources disponibles [Dal03]. Dans la même logique de raisonnement, Braynov *et al.* citent dans [BH03b] les 4 constituants (agent, but, effecteur et mesure de performance) en relation avec l'autonomie dont la performance qui permet de mesurer le succès d'un système artificiel. Cependant la mesure de la performance ne peut pas se limiter à l'estimation du succès, mais elle permet aussi de décider et de choisir le comportement permettant d'atteindre le résultat espéré.

À partir de la présentation des notions liées à l'autonomie, ses modes et niveaux, l'automatisme et l'autonomie comportementale, nous avons souligné que le concept de performance joue un rôle central. Il constitue en effet un critère de choix qui permettra au système artificiel de s'adapter à la mission qu'on lui a fixée, mais aussi aux aléas auxquels il aura à faire face durant son exécution. Ceci apparaît dans [DDDB15] où Teissier traite la question de l'autonomie dans un système de décision. Elle affirme clairement que la

prise de décision est caractérisée par la réactivité (réponse aux changements), l'autonomie (fonctionnement indépendant d'autres agents) et par l'orientation par les buts (capacité à calculer et organiser les décisions).

Nous allons présenter dans la suite ce concept tant dans sa généralité, qu'à travers son utilisation en robotique manufacturière et en robotique mobile.

1.2 Performance

Comme nous l'avons vu dans le paragraphe précédant, la performance est un concept fondamental pour tout système autonome. Plus particulièrement, dans le domaine de la robotique autonome ce concept est plus ou moins défini selon le champ d'application (robotique manufacturière, robotique mobile dans un environnement structuré ou non).

Si la notion de performance est souvent employée en robotique, son emploi est souvent galvaudé et le concept est peu ou pas défini. C'est pourquoi nous allons rappeler dans ce chapitre certaines notions liées au concept de performance et ceux dans différents domaines tels que l'économie et la productique. Parmi ces notions, certaines peuvent être transposées au domaine de la robotique et plus particulièrement à celui de la robotique mobile où leur utilisation manque souvent de rigueur.

1.2.1 Concept de performance

Étymologiquement, l'interprétation latine du terme performance confère plutôt le sens de résultat (implicitement positif). Son acception actuelle relève plus de la terminologie anglo-saxonne qui lui accorde une signification liée autant à l'action qu'au résultat (*to perform*).

Les travaux de synthèse de Bourguignon dans [Bou95] permettent de dégager un ensemble de caractéristiques inhérentes à la performance :

- Elle dépend d'un objectif,
- Elle peut être multidimensionnelle,
- Elle est le résultat de l'action,
- Elle dépend des moyens mis en œuvre,
- Elle peut être subjective car elle est le produit de l'opération qui consiste à rapprocher la réalité (le résultat obtenu) d'un souhait (l'objectif).

Ces caractéristiques montrent explicitement la relation entre les ressources mises en œuvre (les moyens) et le(s) objectif(s) à atteindre.

1.2.1.1 La performance : une entité complexe

En productique la construction et le fonctionnement d'un système de pilotage de la performance repose sur la maîtrise de 4 paramètres : les *objectifs*, les *moyens* d'actions, les *résultats* et la *finalité*. Le système ne sera performant que s'il met en adéquation ces 4 paramètres deux à deux. Cette analyse a conduit à définir les 4 concepts fondamentaux (Figure 1.2) [Bes95 ; Jac90] que sont :

- La *pertinence* qui évalue l'adéquation entre objectifs et moyens. Les moyens engagés sont-ils adaptés aux objectifs fixés ?
- L'*efficience* qui évalue l'adéquation entre moyens et résultats. Les résultats sont-ils suffisants compte tenu des moyens engagés ?
- L'*efficacité* qui caractérise l'articulation entre résultats et objectifs. A-t-on atteint les objectifs fixés ? En est-on éloigné ?
- L'*effectivité* qui évalue l'interaction entre objectifs, moyens et résultats (*in fine*). Le jeu en valait-il la chandelle ?

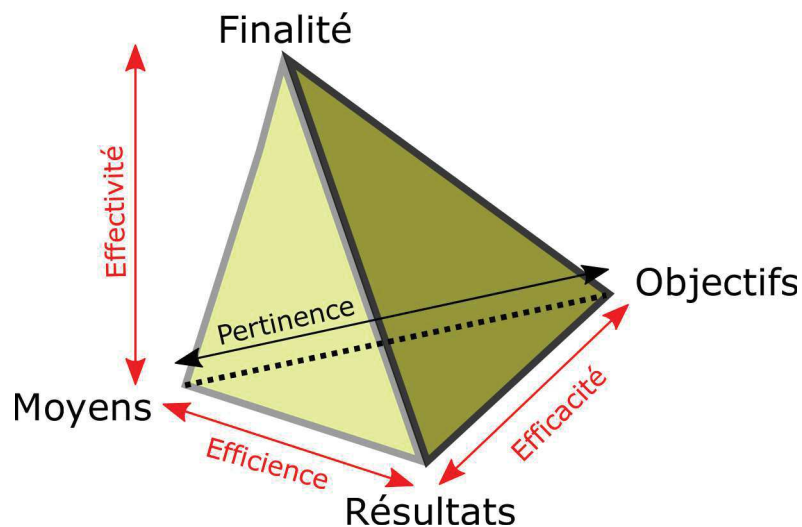


FIGURE 1.2 – Le tétraèdre des performances

Projeter la problématique posée dans ce manuscrit sur les concepts présentés précédemment revient à poser la question suivante : "Quelles *moyens* utiliser pour obtenir les *résultats* respectant les *objectifs* attendus pour atteindre une *finalité* ?".

1.2.1.2 Vues locale et globale

Si la performance s'inscrit dans un cadre global, elle se déploie selon les différents niveaux décisionnels et exécutifs.

- La performance de chaque fonction n'a de sens que si elle permet d'atteindre la performance globale attendue même si bien évidemment, cette dernière ne peut être le résultat de la simple somme de toutes les performances locales.
- Le pilotage doit prendre en compte, non seulement la performance actuelle du système, mais aussi son évolution future (stratégie).

Généralement deux principaux modes de déploiement de la performance sont identifiés [LOR96] :

- Le **mode simple** où la performance globale est la somme des performances locales.

$$Perf_{Globale} = \sum_{Pondérée} Perf_{Locale} \quad (1.1)$$

- Le **mode complexe** ou mode de déploiement "cause-effet", plus explicatif, dans lequel la somme n'est plus algorithmique mais est le résultat d'un enchaînement complexe de causes et d'effets aboutissant à l'identification de leviers d'actions et d'indicateurs.

Il est clair que pour des systèmes relativement sophistiqués tels que les robots mobiles l'évaluation de leur performance va se situer plus en mode complexe qu'en mode simple.

1.2.1.3 La performance : une vision plurielle

Le concept d'évaluation de la performance est polymorphe puisqu'il dépend de nombreux critères :

Quelles performances veut-on évaluer ? Pourquoi ? Pour qui ? Avec quoi ? Etc.

Dans les années 70 les chercheurs ont tenté de dégager un modèle de la performance et d'en dégager un cadre formel. Bien que l'évaluation de la performance s'exprime selon de multiples dimensions, la plupart des grilles d'évaluation privilégient certains aspects au détriment d'autres. Quels que soient les points de vue envisagés, ces modèles présentent des **invariants dimensionnels** en productique : sociétal (méta-politique), financier (stratégique), marchand (tactique), physique (opérationnel), auxquels vient s'ajouter une **dimension temporelle**.

Partant du principe que chaque domaine a ses propres spécificités, il est nécessaire d'identifier les axes invariants et indispensables dans le cadre de la robotique mobile.

1.2.1.4 Les logiques d'évaluation de la performance

Deux logiques d'évaluation de la performance peuvent être dégagées :

- On veut réaliser un audit. L'analyse de la performance se fait donc à partir d'une **post-évaluation** des informations disponibles. C'est le cas de beaucoup de travaux en robotique qui ne sont pas dans une logique de pilotage de la performance.
- On veut mettre en place une **stratégie de pilotage de (par) la performance**. Cela nécessite donc le déploiement de mécanismes permettant de mesurer l'efficacité d'un processus ou d'un système par rapport à une norme. Le déploiement doit alors être exécuté du niveau le plus général au niveau opérationnel. Le travail proposé dans ce manuscrit s'inscrit à ce niveau où le but est de gérer les ressources d'un système robotique afin d'atteindre des performances exigées.

Le pilotage par la performance nécessite d'articuler dynamiquement les indicateurs locaux avec les indicateurs globaux par la mise en place d'un **Système d'Indicateurs de Performance** dont les principes de construction ont été posés dans le domaine de l'entreprise par Kaplan [KN96] et Bitton [Bit90] :

- Les mesures de performance doivent être mises en place au niveau des activités même qui leur donne naissance.
- Les mesures doivent être établies en cohérence avec les objectifs de l'entreprise.
- Les mesures doivent être quantifiables et compréhensibles.
- Les mesures doivent être contrôlables par les personnes à qui elles sont destinées.
- Les mesures doivent être adaptées en fréquence pour pouvoir être validées.

Mêmes si les précédents principes ont été établis pour le contexte de l'entreprise, leurs projections sur le domaine de la robotique semble évidente.

1.2.1.5 Évaluation de la performance

Il est fondamental de distinguer la mesure, de l'estimation et de l'évaluation.

- **Mesure** : C'est la détermination d'une grandeur par comparaison avec une grandeur de même nature prise comme référence [Pru02]. La mesure s'en tient aux effets [Jac90].
- **Estimer** : C'est fournir une valeur centrale à laquelle est associée une plage de confiance. C'est donc l'association d'une mesure et d'une plage de confiance.

- **Évaluer** : L'évaluation délivre une interprétation par rapport à une vision globale ou un cadre de référence dans un contexte donné. Évaluer ne se restreint pas à mesurer. C'est donc attribuer une valeur bonne ou mauvaise, meilleure ou pire [Jac90]. L'évaluation de la performance peut se faire alors à deux niveaux :
 - L'évaluation peut être réalisée *a priori*. Dans ce cas il faut donc **prédire les performances** d'un système futur dans l'objectif de le concevoir ou de modifier l'existant. Après avoir défini les objectifs de référence, on choisit et dimensionne les variables modifiables (ressources) du système opérant, ou du système de pilotage (règles de gestion, procédures, etc.). Les performances du système ainsi modélisées sont alors analysées en **simulation**.
 - L'évaluation *a posteriori* concerne un **système existant** que l'on cherche à piloter par la performance. On peut distinguer là encore 3 phases : La comparaison entre mesures et objectifs, la mise en œuvre d'action pour réduire les écarts observés, l'observation de l'évolution de la mesure.

Ces deux niveaux d'évaluation sont indispensables pour la problématique que nous abordons. L'évaluation *a priori* permet de décider (hors ligne) de la réalisabilité de la mission et celle *a posteriori* assure la gestion (en ligne) du système pour garder le niveau de performance souhaité. L'évaluation et la gestion de la performance d'un système reposent respectivement sur l'information donnée par un **indicateur** de performance et l'action induite par un **inducteur** de performance.

1.2.1.6 Indicateurs et inducteurs de performance

L'association française de gestion industrielle définit ainsi l'Indicateur de Performance (IP) :

a. Indicateur de performance

C'est une donnée **quantifiée** qui **mesure** l'efficacité de tout ou partie d'un processus ou d'un système par rapport à une norme, un plan ou un objectif [Gal].

On peut définir une taxonomie entre indicateurs de performance :

- Les **indicateurs de résultat** : ce sont des indicateurs qui renseignent sur la réalisation d'un objectif. Ils n'agissent pas directement sur l'objet qu'ils mesurent. Ils peuvent être considérés comme statiques pour le pilotage. Il n'existe pas de relations causales entre eux et les variables d'actions du système. Ils relèvent plus du constat que de l'action.

- Les **indicateurs de pilotage (de progrès)** ou **inducteurs de performance** :
Ce sont des indicateurs qui établissent un lien causal entre l'objet de la mesure et la performance. C'est auprès de ces indicateurs qu'il va falloir intervenir pour effectuer des corrections/modifications permettant de piloter le système.

La **période d'échantillonnage** permettant le calcul des indicateurs de performance est une dimension du problème qu'il convient de prendre en compte pour disposer d'informations valides pour la prise de décision.

Une **variable d'action** est nécessairement associée à un indicateur pour le faire évoluer de façon à réduire l'écart entre valeur constatée et objectif assigné.

b. Identification des inducteurs de performance

L'approche d'identification des inducteurs de performance, souvent utilisée dans le domaine de l'entreprise, peut être utilisée pour un système robotique. Ainsi le mot "système" désignera dans la suite aussi bien l'entreprise que le robot. Le système d'indicateurs (Figure 1.4) doit être à l'image d'un modèle causes-effets (Figure 1.3). Dans sa forme la plus pure le modèle est arborescent, cependant, compte tenu de la complexité du système étudié et de l'environnement incertain, une telle structure arborescente est difficile à construire. Il existe souvent des corrélations bouclées pouvant influencer plusieurs objectifs de façon contradictoire [MBC01].

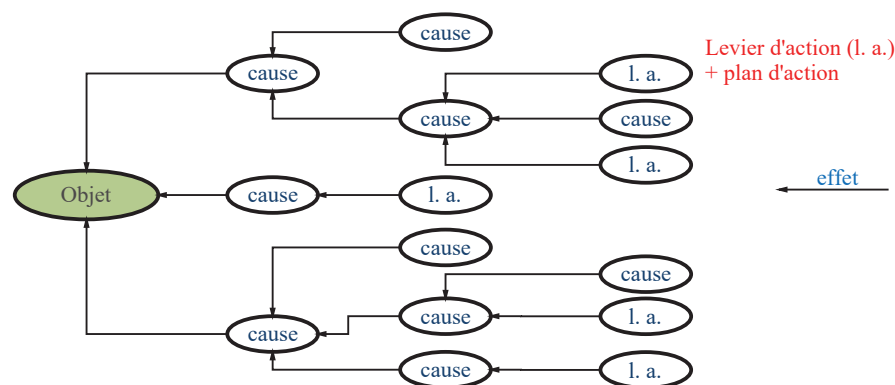


FIGURE 1.3 – Structure causes-effets [MBC01] : l'arborescence indiquant plusieurs niveaux de causalité qui lient un objet aux potentielles actions qui l'influencent.

Pour construire le modèles causes-effets il faut analyser l'enchaînement des activités d'un système bien évidemment en prenant un point de vue s'intéressant aux sorties sur lesquelles on souhaite avoir une influence.

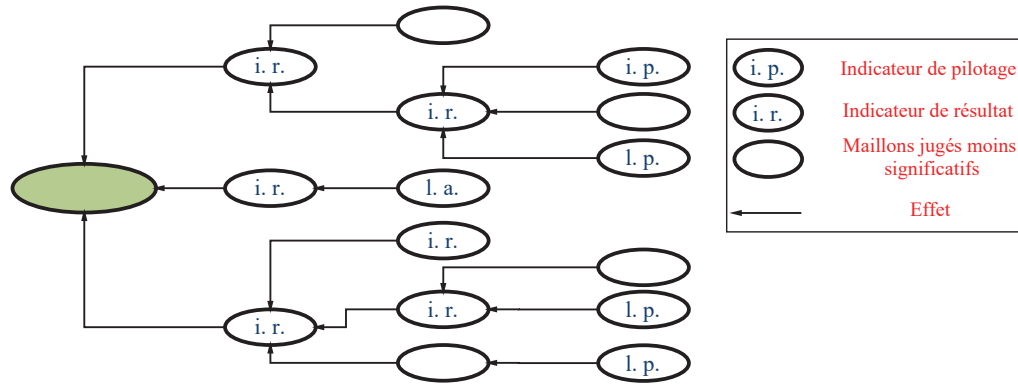


FIGURE 1.4 – Système d'indicateurs correspondant à la structure causes-effets de l'exemple de la Figure 1.3 [MBC01]

L'identification des interactions et des liens de causalité entre les différents éléments pouvant impacter un objectif est donc une étape centrale de la mise en place d'une stratégie de pilotage par la performance. Un certain nombre d'outils peuvent être utilisés :

- **Le diagramme d'Ishikawa** [IL85], est l'outil par excellence permettant d'identifier les relations causes-effets selon 5 points de vue mis en avant dans la méthode 5-M (milieux, méthode, matière, matériel et main d'œuvre). Ce diagramme représente de façon graphique les *causes* aboutissant à un *effet*. Il peut être utilisé comme outil de modération d'un brainstorming et comme outil de visualisation synthétique et de communication des causes identifiées. Il peut être utilisé dans le cadre de recherche des causes d'un problème existant ou d'identification et de gestion des risques lors de la mise en place d'un projet. Cet outil présente cependant une limitation importante puisqu'il est purement qualitatif.
- **Les matrices d'interaction** : Ces matrices de type DSM (Design Structure Matrice) [Bro01], permettent de représenter les interactions entre éléments d'un même domaine. On distingue les DSM statiques, mettant en évidence des relations de dépendance, des DSM temporelles, mettant en évidence des relations de précédence. Elles présentent un intérêt certain pour la formulation mathématique, la modularité, mais présentent des limites lorsque leurs dimensions deviennent trop importantes.
- **Le diagramme d'interaction** est un outil conventionnel largement utilisé dans l'industrie. Il présente l'intérêt d'être graphique et donc facilement compréhensible si les relations exprimées ne dépassent pas un certain seuil de complexité. Par ailleurs, comme dans le cas des diagrammes d'Ishikawa, il ne précise pas la pondération des interactions. Cette information pourrait être introduite arbitrairement en pondérant les arcs de liaison [PBR93].

Maintenant que nous avons défini les indicateurs et inducteurs de performance ainsi que leur technique d'identification, nous allons aborder le concept de pilotage de la performance.

1.2.1.7 Pilotage de la performance

Le concept de pilotage s'applique à des systèmes complexes difficilement maîtrisables. L'entreprise est un système complexe, composée d'un nombre important de ressources (matérielles et humaines) et la présence de perturbations (panne, retard, etc.) réduit la capacité à en anticiper l'évolution [Erm00]. Morlay [Mor98] avance que les 2 concepts clefs du pilotage sont le contrôle et la régulation :

- Le **contrôle** consiste d'une part à identifier les variables du tableau de bord (ensemble de leviers sur lesquels on peut agir) et les plages de fonctionnement associées, permettant de traduire les objectifs visés pour le système. D'autre part il détermine les moyens d'action pouvant impacter les variables du tableau de bord et donc la satisfaction des objectifs.
- La **régulation** vise à maintenir le système dans les limites de fonctionnement que le contrôle a fixé en fonction des écarts observés, à l'aide des variables d'action.

a. La boucle de pilotage

La boucle de rétroaction est l'élément structurel fondamental de la régulation des décisions [FOR80].

Raviart propose dans [Rav99] un schéma générique (Figure 1.5) de pilotage de la performance :

Ce modèle fait apparaître les fonctions de pilotage suivantes :

- Les fonctions de pilotage principales :
 - L'interprétation de l'effet d'une action de pilotage sur le système opérant. Le même comportement peut conduire à des interprétations (résultats) différentes en fonction de l'objectif visé. Le "centre de pilotage" met en œuvre l'activité de renseignement / d'information, qui exerce souvent une influence notable sur le comportement du système de pilotage (l'observation influe le comportement).
 - L'évaluation (Figure 1.6) consiste à estimer l'effet des actions de pilotage potentielles avec l'état actuel du système (sortie de l'interprétation). Cette comparaison influencera certains indicateurs de performance.

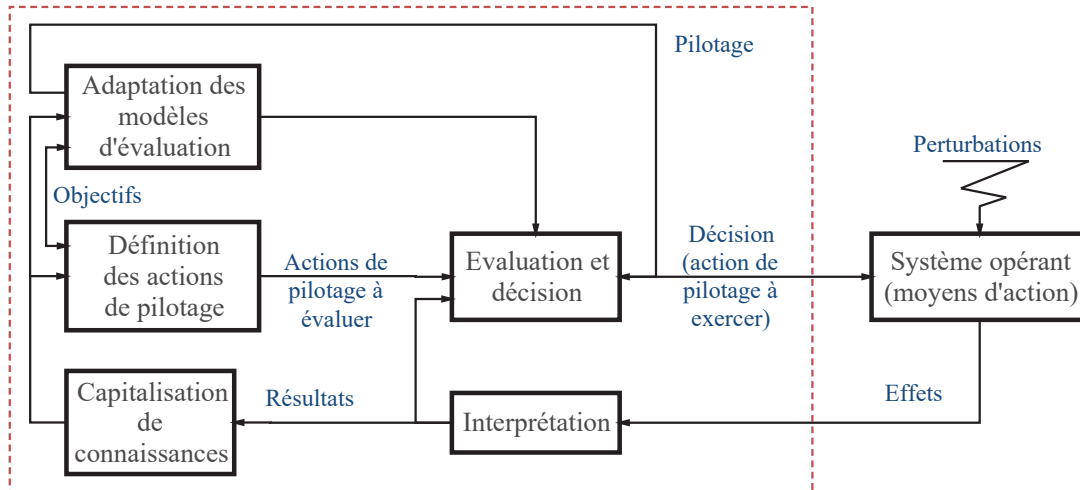


FIGURE 1.5 – Modèle générique du pilotage de la performance proposé dans [Rav99]

- Ces indicateurs de performance guideront le choix / décision (Figure 1.6) de l'action de pilotage à mettre en œuvre sur le système opérant.
- Les fonctions associées :
 - La définition des actions de pilotage potentielles compte tenu des objectifs.
 - La capitalisation des connaissances pour améliorer le choix des actions à mettre en œuvre.
 - L'adaptation des modèles d'évaluation utilisés qu'il faut adapter en fonction des écarts prévus/réalisés observés.

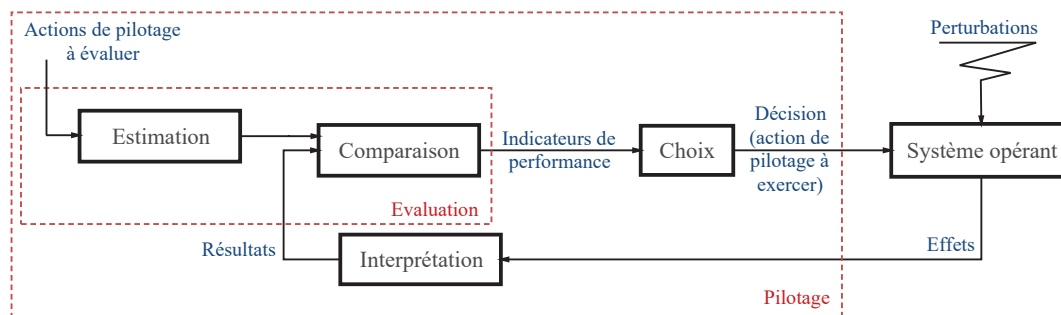


FIGURE 1.6 – Modèle du centre de pilotage [Tah03]

On trouve aussi dans la présentation de la méthodologie GRAI dans [Tah03] un schéma synthétique (Figure 1.7) positionnant clairement la boucle de pilotage par la performance (IP, ressources, etc.).

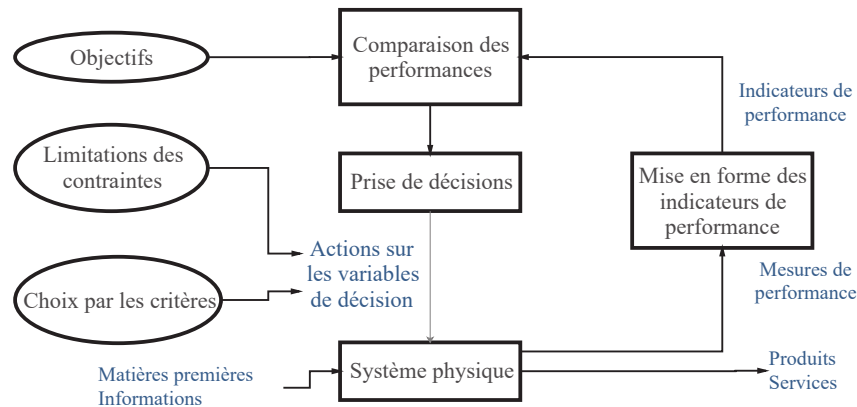


FIGURE 1.7 – Boucle de pilotage [Tah03]

L'analyse précédente a mis en avant certaines notions bien établies dans le contexte de l'entreprise. Nous les retrouvons dans le contexte de la robotique mobile. De plus nous retrouvons les notions d'objectif et de prise de décision caractérisant l'autonomie, ce qui confirme le lien entre autonomie et performance.

b. Commandabilité et observabilité

La **Commandabilité** : elle signifie l'existence d'une séquence d'événements permettant d'atteindre n'importe quel état atteignable à partir d'un état initial quelconque.

Ce concept qualifie le système vis-à-vis de sa commande. Le système sera commandable s'il existe un ensemble de leviers du modèle sur lesquels nous pourrons agir pour piloter le système. Nous appellerons ces leviers des **inducteurs** de performance.

L'**Observabilité** : un système est observable si on dispose des mesures nécessaires pour calculer un contrôle.

Ce concept exprime la capacité d'appréhender et de modéliser (de façon déterministe ou probabiliste) le comportement dynamique d'un système de façon à en prévoir son évolution future. L'observabilité est donc directement reliée à la performance qui permet de suivre l'évolution dynamique du système selon différents points de vues par le biais d'**indicateurs** de performance.

Un système est alors observable si on dispose des mesures nécessaires pour calculer un contrôle. Nous détaillons dans la suite les concepts d'indicateurs (observation) et inducteurs (commande) de performance ainsi que les techniques et mécanismes permettant leur identification.

Nous avons étudié le concept de la performance dans sa généralité bien que la majorité des notions liées à ce concept sont issues du domaine de l'entreprise et de l'industrie. Nous allons maintenant nous focaliser sur l'étude de la performance en robotique manufacturière et en robotique mobile.

1.2.2 Performance en robotique

Dans cette partie nous allons expliciter comment la problématique et les définitions du concept de performance peuvent être projetées du domaine de la modélisation d'entreprise à celui de la robotique industrielle puis mobile.

La performance en robotique est un thème important comme le démontre la création en 1989 [MPM10] d'un comité technique sur l'évaluation de la performance et le benchmarking des systèmes robotiques et automatisés (Technical Committee on Performance evaluation and benchmarking of robotics and automation systems) [Soi]. Les travaux de ce comité soulignent que pour assurer la fiabilité et la robustesse des systèmes robotisés il est crucial de quantifier leur performance à travers des métriques, mesures et méthodes d'évaluation scientifiquement définies. Ils affirment par ailleurs, qu'à part les exigences utilisateurs, il n'existe actuellement aucun standard unanimement accepté pour la mesure qualitative de la performance.

Cependant, force est de constater que, dans l'état de nos connaissances, le concept de performance en robotique, souvent utilisé, n'est pas clairement défini surtout dans le cadre de la robotique mobile.

En effet, si l'on se restreint à ce cadre d'étude, l'analyse des travaux de méta-modélisation récents attestent, de façon étonnante, que ce concept est ignoré dans le projet PROTEUS [Far09] qui définit une ontologie de la robotique, non évoqué dans [PA07] qui s'intéresse aux Domain Specific Languages pour la robotique, et à peine identifié dans la thèse de Goodwin [GW08] portant pourtant sur la conceptualisation d'une mission robotique. Cela démontre à quel point, ce concept, pourtant couramment employé et décliné, reste mal ou peu formalisé en robotique mobile. En revanche l'étude bibliographique démontre que la notion de performance est souvent utilisée pour des robots de manipulation manufacturiers.

1.2.2.1 Robotique manufacturière

Bien évidemment, compte tenu du caractère industriel d'utilisation de cette classe de robots, les instances de standardisation et de normalisation se sont penchées sur la

définition de critères de performance (plutôt que du concept de performance lui-même) [VB90] :

- L’**ANSI/RIA R15.05**¹ *Robot Performance Test Standard* propose les critères suivants : précision du positionnement statique, temps de cycle, temps d’initialisation, répétitivité point-à-point, conformité.

Pour effectuer une comparaison relative entre les robots, une trajectoire et un plan de test standards sont utilisés.

- L’**ISO/DIS 9283**² *Draft International Standard* est une norme similaire mais plus complète que ANSI/RIA R15.05. Elle décrit les performances de tests attendues pour un robot manipulateur à l’intérieur d’un espace de travail cubique à savoir : précision du positionnement, répétitivité, variation de précision, déviation de courbure, conformité statique à 100% de la charge.

D’autres critères ont été proposés dans [Riv87] ou [VB90]. Globalement ces critères relèvent [SM08 ; LE95] de :

- la charge qui est spécifiée en fonction de la distance entre le centre de gravité de la charge et le centre de la surface de montage de l’outil.
- la vélocité (vitesse angulaire et linéaire) est la vitesse de la trajectoire du point central de la surface d’un pince (contact) ou d’un point particulier d’un outil effecteur.
- la précision d’une trajectoire est l’écart en pourcentage entre la vitesse de commande et la moyenne obtenue à partir des mesures répétées sur un horizon de temps spécifique.
- la répétitivité (1 à 2 mm à quelques μm) est le degré de concordance des vitesses obtenues dans les mêmes conditions.
- la variabilité est l’écart maximal de la vitesse au cours d’une reproduction d’une commande de vitesse.
- l’accélération (jusqu’à 100 G [CNP13]) peut être l’accélération individuelle par axe ou celle de la trajectoire de l’outil/effecteur (plus important pour l’utilisateur).
- temps de positionnement qui est le temps mis par le robot pour se déplacer suivant une trajectoire prédéfinie entre deux points stationnaires.
- la résolution qui est le plus petit incrément de position ($< \text{mm}$)).
- l’espace de travail est l’ensemble des poses (dans le repère de référence) pouvant être atteintes par le l’outil.

1. <http://standards.globalspec.com/std/307040/ansi-ria-r15-05-2>

2. http://www.iso.org/iso/fr/catalogue_detail.htm?csnumber=22244

Ces critères sont en général déterminés lors de séquences de tests et donc en dehors du fonctionnement en production.

Des critères de performance plus spécifiques ont été définis dans le cadre de la dextérité [SC82] et de la manipulation [AP08]. Ils cherchent à évaluer la capacité du robot à se déplacer et à appliquer des forces dans une direction arbitraire à travers des critères dynamiques locaux [BK05] ou globaux [GA88]. Ils ont été complétés par un ensemble de mesures cherchant à caractériser les capacités dynamiques d'un robot manipulateur à travers l'accélération radiale [GK88], la performance élastostatique (rigidité du manipulateur) [GD93], et élastodynamique en présence de vibrations [Mei01].

La robotique industrielle a donc largement abordé la question de la performance à travers la définition d'un grand nombre de critères de performance standards. Cependant, ce contexte est très particulier au regard de l'environnement de travail limité, connu et statique (structuré), de la répétitivité des tâches, de la localisation relativement précise, de l'énergie "illimitée" dont on dispose, etc.

Intéressons-nous maintenant aux travaux abordant la notion de performance en robotique mobile.

1.2.2.2 Robotique mobile

En premier lieu, il est important de noter l'analogie entre un processus industriel et une mission robotique. Un processus industriel est composé d'un ensemble d'activités exécutées séquentiellement ou concurremment (dans le temps) par des ressources humaines, logicielles et/ou matérielles. A ce processus est associé un/des objectif(s). L'organisation choisie ainsi que les ressources affectées induisent une performance tant au niveau de chacune des activités (locale) qu'au niveau du processus lui-même (globale), performance pouvant être déclinées selon différents points de vue.

De la même façon une mission robotique peut être déclinée en un ensemble de tâches robotiques s'exécutant séquentiellement ou concurremment, pouvant être réalisées par le robot en totale autonomie, en collaboration avec d'autres robots (flottes) où avec des opérateurs humains. Il faut par ailleurs remarquer qu'une entité robotique physique peut intrinsèquement supporter plusieurs tâches robotiques (décomposition hiérarchique), un robot pouvant se déplacer tout en exerçant une activité à l'aide d'un bras manipulateur, par exemple, où par l'exploitation de la redondance structurelle, différentes tâches hiérarchiques lui sont attribuées. Bien évidemment une mission robotique est déployée de façon à atteindre des objectifs assignés de la façon la plus performante possible.

Toutefois, une mission robotique est exécutée dans des conditions spécifiques par rapport à la robotique manufacturière. Généralement l'environnement dans lequel évolue(ent) le(s) robot(s) est dynamique, complexe, peu (ou non) connu tout en disposant d'une "intelligence embarquée" limitée. La localisation ainsi que la perception est fortement dépendante de la technologie des capteurs utilisés et de la nature de l'environnement. De plus, l'énergie est limitée et le déroulement de la mission n'est pas forcément conforme à celui prévu.

Un certain nombre de métriques ont été proposées dans la littérature. Elles sont souvent liées au besoin d'évaluation de l'utilisateur. Néanmoins ces critères manquent de formalisme et standardisation. Caballos *et al.* mettent en avant dans [CVO10], comme dans [Won+02] que malgré le large éventail de travaux relevant de la navigation de robots mobiles, bien peu d'études s'intéressent à la performance des algorithmes utilisés, ce qui est fort dommageable dans une perspective d'implémentation industrielle pour pouvoir mettre en œuvre les techniques les plus performantes. Ils insistent aussi sur l'importance de la définition de procédures de test et de benchmarks [BHP08] permettant de pouvoir comparer objectivement la performance des algorithmes de même nature.

Dans [BHP08] plusieurs auteurs identifient des critères objectifs de performance associés à différentes classes de tâches robotiques. Les simulations sont très souvent mises en avant pour évaluer les performances en raison de la facilité de mise en œuvre et de la reproductibilité des conditions expérimentales, même si l'on ne peut s'affranchir de la réalité de terrain qui consiste à rechercher dans l'espace des connaissances *a priori* et en construction, les localisations possibles qui engendrent une erreur faible entre les mesures courantes (espace sensoriel) et leur estimation dans la carte.

Nous allons maintenant balayer un ensemble de métriques de la performance proposées en robotique mobile dans la littérature.

a. Contrôle du déplacement, évitement d'obstacles et sécurité

Un des aspects clef de la performance des robots mobiles est le suivi de trajectoire. Pour l'évaluer des critères comme la vitesse, la robustesse et la précision sont étudiés. L'erreur entre la commande de référence et la réponse du robot est usuellement utilisée par les automaticiens. Ils abordent la question de la performance *in fine*, de façon à valider expérimentalement leurs approches de contrôle, mais sur une trajectoire unique et pour des missions courtes. Ceci ne permet donc pas d'en trouver les limitations et de les comparer avec d'autres contrôleurs (Marques L. dans [BHP08]). Akanyeti *et al.* utilisent dans [Aka+10] des métriques (nombre de passages réussis, distance et angle par rapport à une porte, etc.) pour évaluer les algorithmes de commande d'un robot mobile ainsi que les modèles de perception lui permettant de passer par des zones étroites.

D'après Minguez J. dans [BHP08], deux points de vue complémentaires sont à considérer. D'une part la génération d'une trajectoire exempte de toute collision et, d'autre part, l'évitement réactif d'obstacles détectés au cours du déplacement. La performance des algorithmes déployés est fortement dépendante du type de robot (dimensions, cinématique, dynamique), du type de capteurs déployés, du scénario envisagé (dynamisme de l'environnement : nombre d'obstacles en mouvement, vitesse, accélération). Cela rend les critères de performance fortement dépendants des conditions expérimentales, qui doivent être caractérisées en termes de : densité du scénario, complexité, confinement, encombrement, etc. [RM06]. Les critères de performance identifiés [SPB02a ; Hol+05 ; Kui06 ; Mon+10] sont alors :

- Nombre de missions accomplies avec succès,
- Longueur du chemin,
- Temps mis pour réaliser la tâche,
- Vitesse moyenne de déplacement pour réaliser la tâche,
- Nombre de collisions par mission, par distance, par durée,
- Nombre de passages étroits franchis avec succès.

Ces critères montrent l'importance de la notion de scénario d'une mission et l'environnement dans lequel elle est exécutée pour analyser la performance d'un robot mobile.

La performance sécuritaire et la qualité de la trajectoire sont étroitement liées à la mobilité du robot. Des métriques comme la longueur, la période de contrôle et le lissage sont proposés dans [Tre+10] et testées en simulation avec différents contrôleurs.

Dans [CVO10] les auteurs s'intéressent également aux métriques de performance utilisées en robotique mobile pour évaluer la performance des algorithmes de navigation et d'évitement d'obstacle. Ils identifient les métriques suivantes inspirées de Minguez dans [BHP08], et y ajoutent une relation d'ordre classant les axes de performance par importance décroissante :

- Sécuritaire : le nombre de collisions par mission, par distance, par durée. La distance de sécurité aux obstacles : minimum et médian.
- Longueur de la trajectoire jusqu'à l'objectif : la longueur du chemin accompli pour réaliser la mission.
- Qualité de la trajectoire : lissage de la trajectoire et capacité de franchissement de passages étroits.

À ces critères s'ajoutent le nombre de missions réalisées avec succès ainsi que leur durée.

[TC 12] présente une étude très détaillée des critères de performance de plateformes mobiles intelligentes domestiques, menée par le groupe de travail de l'organisme de normalisation *International Standards and Conformity*. Les principaux critères identifiés sont :

- **Positionnement** : la précision d'une pose (x,y,θ) et d'une commande de parcours d'une distance, la répétitivité d'un chemin et la déviation angulaire.
- **Capacités du robot** :
 - Mouvement : Accélération / décélération,
 - Capacité à surmonter des obstacles sur/sous la surface de travail,
 - Durée d'utilisation après charge maximale sous conditions nominales d'utilisation,
 - Charge embarquée maximale sous conditions nominales d'utilisation,
 - Consommation à l'arrêt (standby).
- **Intelligence embarquée** :
 - Capacité à revenir à sa station d'accueil,
 - Évitement d'obstacle,
 - Planification de chemin entre une position courante et finale,
 - Évitement de chute,
 - Résistance au *kidnapping* (localisation avec position initiale inconnue).

Nous pouvons remarquer que dans les différents travaux qui s'intéressent à la performance du robot en termes de contrôle évoquent implicitement ou explicitement la question sécuritaire.

b. Localisation

b.1. Odométrie

L'odométrie procure une méthode simple de localisation qui, malheureusement, est entachée par de nombreuses erreurs systématiques (Diamètres de roues inégaux, roues mal alignées, résolution des encodeurs, fréquence d'échantillonnage des encodeurs) et non systématiques (sols inégaux, dérapage). De nombreux travaux se sont intéressés à l'estimation de la dérive odométrique [CLTC95] [BF96] [DCC03] [Gof10]. Pour approfondir l'étude on trouvera dans [SN04] une formulation détaillée et générale (ou presque) de l'erreur de localisation odométrique pour les robots à conduite différentielle comme le Pioneer-3DX.

b.2. SLAM : Simultaneous Localization And Mapping

Neira *et al.* dans [BHP08] ont établi des critères de performance de SLAM.

- La précision de la carte : exemple erreur par rapport à la réalité (métrique Hausdorff).
- La mise à l'échelle : quel est l'accroissement de la charge de calcul en fonction de la surface couverte, du nombre d'amers identifiés et/ou nécessaires en fonction de la longueur de trajectoire.
- La "confiance" : si une estimation de l'erreur est calculée, celle-ci est-elle réaliste, optimiste, pessimiste ?

Nous trouvons quasiment les mêmes critères dans [SPB02b] pour évaluer la localisation issue du SLAM.

c. La manipulation

Cette tâche, commune évidemment en robotique de manipulation industrielle, peut aussi être rencontrée dans les applications de robotique mobile lorsque le robot doit interagir avec des objets pour modifier leur position, les transporter ou les utiliser en tant qu'outils. Le plus souvent on fait appel à un bras robotisé auquel est adjoind un outil préhenseur (doigts) qui assure le contact avec l'objet.

Cette tâche est complexe car elle inclut le contrôle de la pince (force exercée, points de contact), la planification de la saisie (localisation des points de contact pour produire l'effet désiré), l'analyse de la saisie (propriétés d'une configuration de saisie). Les critères de performance de manipulation peuvent être complétés par ceux identifiés dans le paragraphe précédent consacré à la robotique de manipulation. Ils sont aussi dépendant des conditions expérimentales : processeur, mémoire, capteurs, nature de la pince, et environnementales : lumière (qualité de vision), présence d'obstacles, type d'objets manipulés. Les critères identifiés dans [BHP08] sont :

- La qualité de saisie,
- La stabilité de saisie,
- Les configurations de saisie,
- Le temps de calcul.

d. L'asservissement visuel

Le *Visual Servoing* abordé par Cervera E. *et al.* dans [BHP08]) est une application pluridisciplinaire associant la vision, la robotique, la cinématique et la dynamique, le contrôle et le temps réel, où les mécanismes de vision sont utilisés dans la boucle de contrôle. Là encore la performance des algorithmes est fortement dépendante des conditions expérimentales (processeur, mémoire) et environnementales (éclairage de l'environnement). Les indicateurs de performance communément rencontrés sont :

- Le temps de convergence de l'algorithme,
- La trajectoire 2-D,
- La trajectoire 3-D,
- Le temps de calcul,
- L'erreur de positionnement après convergence,
- La stabilité vis-à-vis du bruit dans l'image, des erreurs de modélisation (objet, caméra, robot), des paramètres de contrôle,
- La robustesse vis-à-vis du bruit dans l'image, des erreurs de modélisation (objet, caméra, robot), des paramètres de contrôle.

e. Autonomie et tâches cognitives

Ce domaine relève clairement de l'intelligence artificielle proprement dite puisqu'ici il faut être capable d'évaluer et de comparer les performances relevant de la représentation des connaissances, des modèles de l'environnement, de la planification, de l'apprentissage et de l'adaptation, du raisonnement [Alb02 ; CG03 ; BHP08 ; MM05]. Dans ce contexte, l'intelligence peut être définie comme étant "la capacité à agir de façon appropriée dans un environnement incertain, où une action appropriée accroît la probabilité de succès de réalisation des objectifs comportementaux définis" [Alb91]. Les quelques critères identifiés sont :

- Le coût de calcul des algorithmes de recherche,
- Le temps de convergence de ces algorithmes,
- Le temps d'apprentissage en fonction de l'efficacité de réalisation d'une tâche,
- Stabilité et robustesse des algorithmes.

Les robots mobiles sont souvent supervisés par un opérateur. Ainsi, les auteurs s'intéressent dans [CG03] à la performance de systèmes robotisés composés d'hommes et de

robots. Ils définissent deux critères de performance pour qualifier la qualité de l'interaction de ces deux entités :

- La **tolérance à l'inattention** (*neglected tolerance*) qui cherche à évaluer comment la performance attendue du robot évolue lorsque l'opérateur le néglige (Figure 1.8). De façon générale, plus l'opérateur le néglige, plus la performance du robot diminue, cependant cette décroissance dépend du schéma d'interaction déployé (mode d'autonomie : téléopération, télé-programmation, autonomie totale).

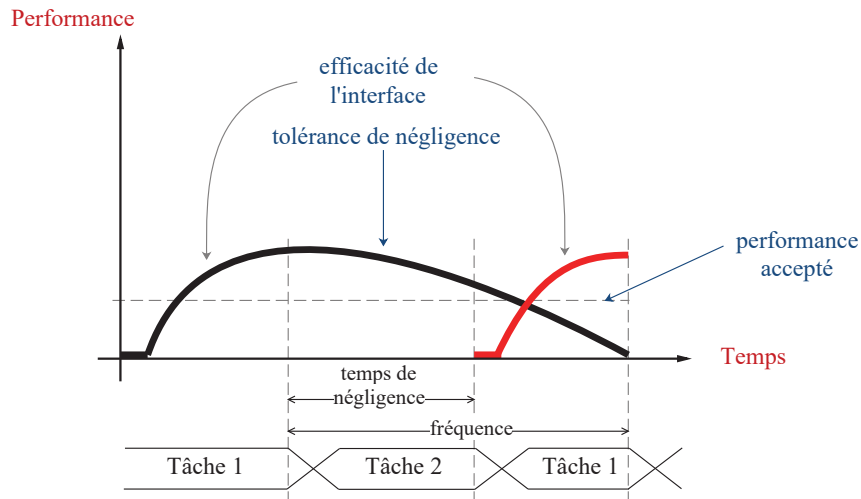


FIGURE 1.8 – Performance et tolérance à l'inattention

- L'**efficacité de l'interface** : l'interface entre l'opérateur et le robot est composée de deux boucles et 3 agents (opérateur, interface, robot) : la qualité de l'interface affecte le temps que mettra l'opérateur pour appréhender une situation (*situation awareness*), choisir une action, déterminer les ordres à envoyer au robot et les lui adresser.
- La **tolérance à la complexité** de l'environnement (*complexity tolerant*) : ce critère de performance reste aussi lié à la complexité de l'environnement dans lequel évolue le robot puisque la performance de ce dernier diminuera avec l'augmentation de sa complexité. Cependant certains schémas d'interaction peuvent se révéler plus ou moins sensibles à l'évolution de la complexité de l'environnement d'où la définition du critère de tolérance à la complexité. Toutefois il n'est pas évident de mesurer la complexité de l'environnement puisqu'elle dépend de la tâche à réaliser et des capacités sensorielles embarquées. Pour une tâche de navigation on peut considérer par exemple le nombre de croisements d'intersections par zone ou le nombre d'obstacles par zone.

Les auteurs de [CG03] postulent que la performance des systèmes semi-autonomes diminue lorsque l'opérateur n'interagit pas avec le robot et que la complexité de l'environnement augmente. Ainsi pour maintenir un certain niveau de performance, il faut ajuster la fréquence et la durée d'interaction de l'opérateur avec le robot en fonction de la complexité de l'environnement (Figure 1.8).

Clough propose une métrique tridimensionnelle permettant de mesurer l'autonomie [Clo02]. Les niveaux s'incrémentent exponentiellement suivant 3 axes (perception, communication/coopération et décision/analyse). Le niveau global d'autonomie est représenté par le triangle dont les sommets sont les valeurs de chaque métrique (Figure 1.9).

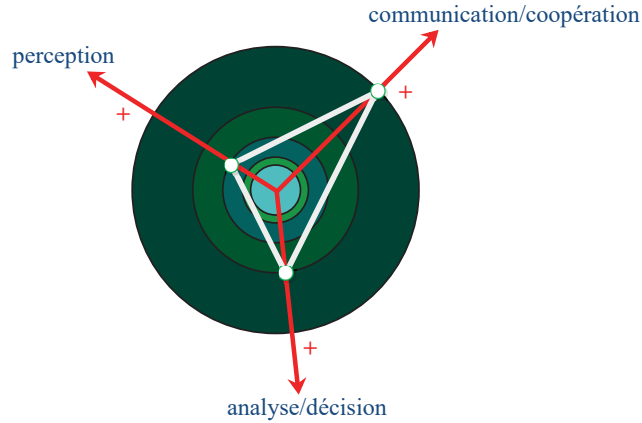


FIGURE 1.9 – Schéma d'illustration de la mesure d'autonomie proposée par [Clo02]

Bien que cette métrique permet d'interpréter facilement un degré d'autonomie d'un système robotique, elle reste purement qualitative et subjective. De plus, plusieurs configurations de triangles peuvent avoir la même note globale (niveau de chaque sommet) tout en ayant pratiquement des niveaux d'autonomie distincts.

f. Service rendu

Wong *et al.* se focalisent dans [Won+02] sur l'évaluation de performance associée à la surface de couverture par des robots mobiles. Ils introduisent un observateur externe (caméra) et se focalisent sur les métriques de performance associées aux tâches de couverture d'une surface bien utiles dans de nombreuses applications robotiques comme, le nettoyage [Pal+10], l'enrobage, le fourragement. Ils définissent deux critères de performance de couverture :

- Le pourcentage de couverture C . Ce critère est établi à l'aide d'une caméra externe.
- L'efficacité de couverture E qui prend en compte la couverture C , la distance parcourue D_a , et la distance optimale D_{min} . $E = D_a / (D_{min} \cdot C)$.

Pour s'assurer de l'efficacité des robots d'exploration, certains métriques ont été définies dans [Sch+09c] pour évaluer en ligne la performance de :

- **Tâches robotiques** : (temps de conduite, temps d'exécution, distance achevée, distance planifiée, pourcentage de distance parcourue). Les performances observées sont comparées avec celles estimées pour savoir si la mission se déroule conformément à ce qui a été prévu.
- **Capteurs (Lidar)** : la performance de mesure du Lidar utilisé pour la cartographie 3D est évaluée par des métriques prédéfinies (progrès du panorama, durée d'exécution, pourcentage du balayage effectué, etc.).
- **Qualité de communication** : mesure la qualité de communication en comptabilisant la taille des données perdues et leur nombre.

Parmi les métriques proposées dans [SAM05], quelques unes ont été utilisées dans [Sch+09a; SMF10] pour mesurer en temps réel la performance des robots faisant de l'exploration spatiale. Ces mesures sont par la suite interprétées pour évaluer la productivité du robot ainsi que ses moyennes de panne comme le temps moyen pour intervenir et le temps moyen entre deux interventions. Dans [Sel+06a], la performance en termes de moyenne de temps pour compléter une tâche, taux de succès et charge de travail sont des indicateurs permettant d'adapter le mode d'autonomie d'un système coopératif Homme-Robot. Carlson *et al.* utilisent ces métriques dans [CM05b] pour évaluer l'efficacité des robots mobiles vis-à-vis des fautes sur les ressources réelles et ceci pour des missions réelles.

La performance des systèmes multi-robots en terme d'efficacité, apprentissage et robustesse par la tolérance aux fautes est évaluée dans [KP07]. Une mesure de la performance du système robotique est aussi proposée. On retrouve des métriques relevant de la même logique que pour les tâches robotiques dans [Sch+09c]. La performance est aussi utilisée pour agir sur un système multi-robot en allouant des tâches aux différents robots constituant la flottille pour une tâche de "*patrolling*" [PCW13]. Le temps de rafraîchissement est alors la métrique principale utilisée. Elle correspond au temps entre deux visites d'un même endroit.

1.2.3 Conclusion

Cette section a permis de présenter le concept de performance en partant des définitions générales issues de la productique et leur projection dans le domaine de la robotique industrielle jusqu'à la robotique mobile. Différents termes liés à la robotique ont été définis. Beaucoup de critères et indicateurs de performance existent dans la littérature. Nous avons remarqué un usage plus normatif dans la définition de la performance en robotique

manufacturière ainsi qu’une standardisation des différentes métriques. Quant à la robotique mobile, les métriques sont généralement spécifiques aux applications pour lesquelles a été développé le robot. Le but est souvent l’évaluation de ce système (algorithmes, mécanisme, etc.) ou sa comparaison avec d’autres systèmes. Un besoin de formalisation et de standardisation du concept de performance en robotique mobile s’impose. Le travail dans [TC 12] démontre l’importance de la performance pour les organismes de normalisation. On retrouve des critères relatifs à la qualité de la trajectoire, et à l’intelligence embarquée. On voit aussi apparaître pour la première fois des critères relatifs aux capacités énergétiques. Cependant, ces critères ne couvrent pas l’ensemble des applications robotiques mobiles.

L’objectif principal de cette thèse est la garantie de performance. Ainsi, l’aspect estimation/prévision/prédiction est primordial pour atteindre cette finalité mais rares sont les travaux qui cherchent à estimer *a priori* la performance dans le but de la piloter. [CG03] propose une formulation générique de la notion de performance évaluée entre -1 et 1, ainsi que la notion d’évaluation continue, discontinue, estimée et instantanée de la performance. Cependant, cette estimation est limitée à l’aspect interaction homme-robot en relation avec la complexité de l’environnement.

Nous visons dans cette thèse la garantie de performance par la gestion des ressources du système robotique durant une mission robotique. Ceci implique impérativement un processus de prise de décision sous des contraintes de performance. Nous allons donc explorer dans la suite la problématique de décision multicritère en robotique. Différentes techniques provenant d’autres domaines seront aussi évoquées.

1.3 Décision multicritère

Les problèmes de décision et plus particulièrement de choix de ressources sont évidemment très répandus. Par exemple [Med11] analyse comment utiliser équitablement des *ressources partagées* en différents usages (choix de fréquences radio, approvisionnement en eau potable, occupation mémoire, ressources physiques, etc.). De nombreux travaux s’intéressent au choix des ressources propres aux systèmes dans le but d’optimiser ou maximiser leur rendement (paramétrisation des modules d’un robot pour optimiser ses performances [Roh+10], de reconfiguration de capteurs pour assurer la sécurité d’un ensemble de cibles [Gro15], ou d’adaptation du niveau d’autonomie par l’allocation des tâches entre homme et robots [Sel+06b]).

Dans le cadre de cette thèse, nous avons vu que l’autonomie impose de faire des choix (décider) sous des contraintes de performance. Ce problème de décision en robotique est

nécessairement multicritère comme l'affirme Sukhatme [Suk97]. Nous allons tout d'abord analyser la relation entre la mission et la problématique de décision multicritère afin de caractériser ce problème et explorer par la suite les différentes techniques de résolution.

1.3.1 Mission et décision multicritère

La problématique de décision multicritère existe dans différents domaines tels qu'en économie ou l'industrie mais elle n'est pas souvent abordée en robotique pour des missions autonomes où le problème de décision peut être très complexe.

Ainsi, le changement d'environnement, la défaillance du système étudié ou de certains de ses composants ou les dérives de performance peuvent perturber son fonctionnement et nécessitent forcément une reconfiguration du système pour optimiser, maximiser ou garantir un certain niveau de performance. Le succès des missions (en général) dans des conditions d'incertitude nécessite, entre autres, le développement d'approches de gestion de ressources assurant un certain degré d'adaptabilité.

Dans ce cadre, Groot propose dans [Gro15] une approche d'aide à la décision pour gérer au niveau opérationnel un ensemble de capteurs reconfigurables utilisés par la défense. Le mécanisme de décision est guidé par le critère de succès de la mission (*End-User mission*). Le problème mathématique est formulé sous la forme d'un problème de type "sac à dos" bien connu pour être NP-complet. Selon Groot, la mission est considérée comme pleinement réussie si tout ce qui est prévu se réalise et tout ce qu'il ne devait se produire ne se produit pas. Tous les objectifs sont alors atteints. Une fonction d'utilité $U(t_e)$ est définie pour quantifier le succès d'une mission ayant Z objectifs. $u_Z(t_e)$ est la fonction d'utilité du dernier objectif en fin de la mission t_e (Eq. 1.2). La complexité de la mise en œuvre de cette formulation réside dans la définition de u_Z . Nous pouvons remarquer aussi que la notion de scénario de mission représentant une suite d'objectifs est clairement adoptée.

$$U(t_e) = \sum_{z=1}^Z u_Z(t_e) \quad (1.2)$$

Cependant, la fonction d'utilité ne peut connaître la performance au cours de la mission et ne permet que d'évaluer le succès final. Cela nécessite la définition d'une fonction d'utilité estimée $U_E(t_e, t)$ qui permet de prendre des décisions en considérant une prédiction probabiliste $P(S_Z, t_e, t)$ qui est fonction de l'état actuel (à t) et celui prévu en fin de mission (à t_e) (1.3). S_Z est une variable binaire indiquant si l'objectif, dans le cadre de la mission envisagée, est atteint ou non.

$$U_E(t_e, t) = \sum_{z=1}^Z u_Z(t_e, t) P(S_Z, t_e, t) \quad (1.3)$$

La figure 1.10 montre le processus d'allocation de ressources sous la contrainte de maximisation du succès d'une mission. Cette démarche révèle l'importance de la phase de définition (planification) de la mission ainsi que des objectifs. Cette étape est indispensable pour réaliser la phase d'estimation. La solution d'allocation déterminée est automatiquement adaptée aux possibles perturbations (erreurs de modélisation de l'environnement, pannes de ressources, changement d'objectifs) par une réallocation optimisée des ressources.

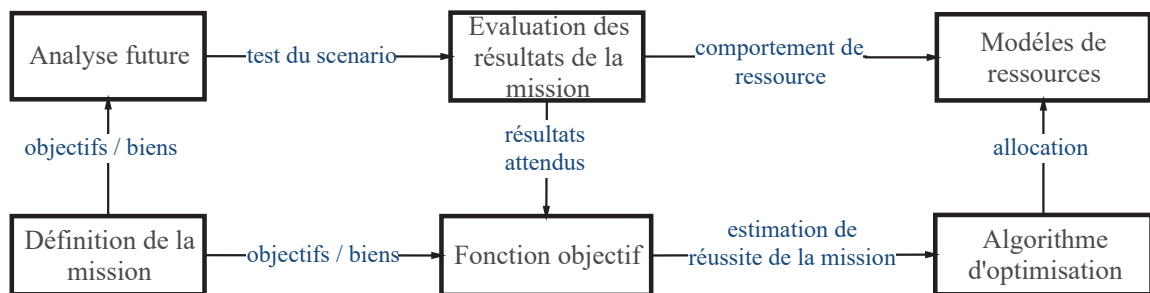


FIGURE 1.10 – Optimisation basée mission

La problématique de décision multicritère a été étudiée aussi en robotique mobile mais dans un contexte différent de Groot par Sukhatme. Dans [Suk97], il propose un système de fonctions d'évaluation multicritère permettant de comparer les performances attendues de robots mobiles du point de vue de la conception. Une définition "claire" des objectifs de la mission est indispensable. La notion de scénario est à nouveau prépondérante pour la prise de décision. L'aspect mission/tâche permet de construire le modèle de simulation pour évaluer les performances attendues (Figure 1.11).

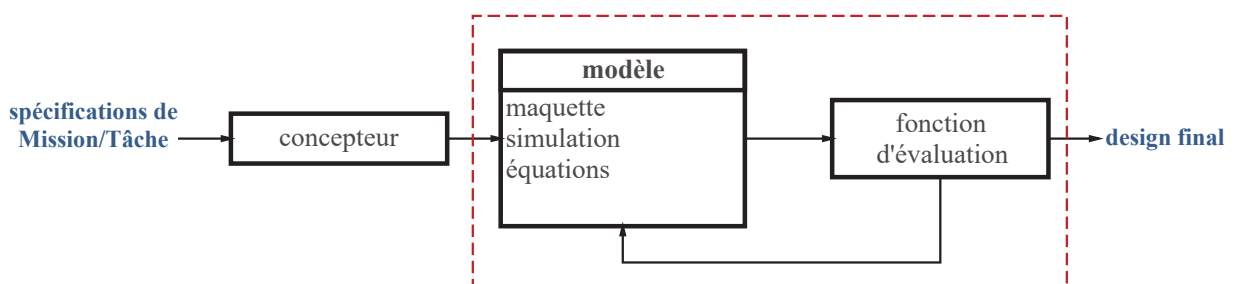


FIGURE 1.11 – Un cycle de conception typique

Cette démarche, bien qu'elle soit hors ligne et en simulation, met en lumière des notions indispensables pour la prise de décision comme la conception d'un scénario de mission décrivant ses objectifs et ses spécifications, ou la définition de fonctions d'évaluation (dans

le cas multicritère) permettant de classer les différentes conceptions possibles pour enfin déterminer la plus adaptée au scénario envisagé.

Une autre forme de décision est présentée dans [Sel+06a]. Elle permet une adaptation dynamique du niveau d'autonomie d'un système multi-agent (robot, homme) guidée par la valeur de certaines métriques de performance telles que l'efficacité, la robustesse et la probabilité de succès.

La décision a été définie par [Sim60] comme étant un processus comportant des phases de définition du but ou des **objectifs**, d'identification des **contraintes** et d'identification des **options**.

Dans le schéma entité-relation d'Ullman [Ull02] (Figure 1.12), la décision est un problème de choix entre différentes **alternatives**. Il définit une alternative comme une option générée pour répondre à un problème particulier. Le problème est défini et délimité par des critères, éventuellement décomposables et hiérarchisables. Un critère est une règle de jugement permettant de juger du niveau de satisfaction des alternatives. Les connaissances possibles sur les solutions existantes et les résultats d'analyse sont utilisés pour classer ces alternatives.

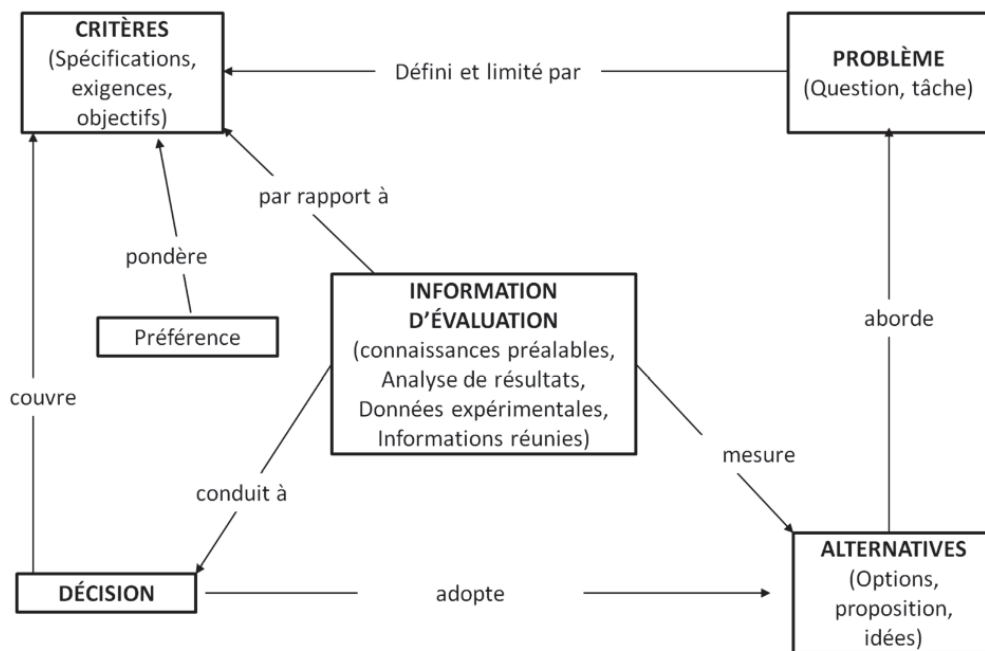


FIGURE 1.12 – Processus de décision selon Ullman (figure tirée de [Ull02])

La définition d'une "fonction objectif" peut être problématique en raison de la variété des critères à prendre en compte. Cependant l'analyse multicritère peut apporter des réponses même en présence d'incertitude ou d'imprécision. L'évaluation d'alternatives, relativement à un critère, peut se traduire par l'attribution d'un niveau interprété selon

des échelonnements ou des échelles [Ste46] [Lo13] de satisfaction.

- Échelle nominale reposant sur une relation d'équivalence (correct/incorrect).
- Échelle ordinale reposant sur une relation d'ordre généralement non réflexive. La distance ou les différences entre ces valeurs ne peuvent être interprétées.
- Échelle numérique cardinale où les valeurs sont définies à une transformation affine croissante près (échelle ordinale) ou linéaire près (échelle de ratio).

Lorsque plusieurs critères doivent être pris en compte, les grandeurs à examiner doivent être converties en une échelle unique qui le plus souvent est une échelle d'utilité [Fis68]. Cette fonction d'utilité fait correspondre à une valeur de grandeur à examiner une valeur d'utilité, réelle dans l'intervalle $[0,1]$ mais qui est difficile à construire [Fül05].

A travers les différents travaux présentés dans cette partie, nous pouvons constater que le problème de décision multicritère dans le cadre de mission (robotique en particulier) est lié à un certain nombre de concepts :

- Une phase de **planification** de mission/tâches est une étape indispensable dans le processus de formalisation du problème. Elle permet d'établir le **scénario** de la mission visée à partir duquel il est possible de calculer le nombre de choix possibles (complexité du problème).
- La deuxième dimension est le nombre d'**actions/alternatives/options** possibles. Une **fonction d'estimation/utilité** permet de classer et d'ordonner les précédentes alternatives. Souvent le problème est formalisé sous forme de problème **sac à dos**.

Différentes techniques permettent de guider le mécanisme de prise de décision tout en réduisant la complexité du problème. Elles sont étudiées maintenant.

1.3.2 Techniques de résolution

1.3.2.1 Méthodes probabilistes

La prise de décision par approche probabiliste est adoptée dans différents travaux pour résoudre des problématiques d'une grande complexité. Une approche d'optimisation de la performance, pour des systèmes robotiques complexes, basée sur l'apprentissage d'un modèle (réseau bayésien) de métriques interdépendantes est proposée dans [Roh+10]. Ces modèles obtenus par apprentissage permettent de palier à la complexité de modélisation analytique pour des systèmes complexes. Ils permettent d'estimer la performance et la paramétrisation des tâches durant l'exécution d'une mission robotique sous un ensemble

d'objectifs. D'une part, la phase d'apprentissage des réseaux bayésiens est faite hors ligne. D'autre part, l'optimisation en ligne se fait sur deux niveaux. Une optimisation globale concerne le planification des tâches suivie d'une optimisation de plus bas niveau assurée par la paramétrisation des tâches exécutées.

Dans [Sel+06a], un arbre de décision permet de représenter les enchainements de commutations robot autonome/robot télé-opéré permettant de réaliser une tâche robotique avec succès. La durée estimée jusqu'à l'échec est calculée par des modèles probabilistes. Le système de décision est adapté dynamiquement en fonction de la performance de l'homme et de sa disponibilité. Bien évidemment, la contrainte globale est la minimisation du temps d'exécution d'une tâche (Figure 1.13).

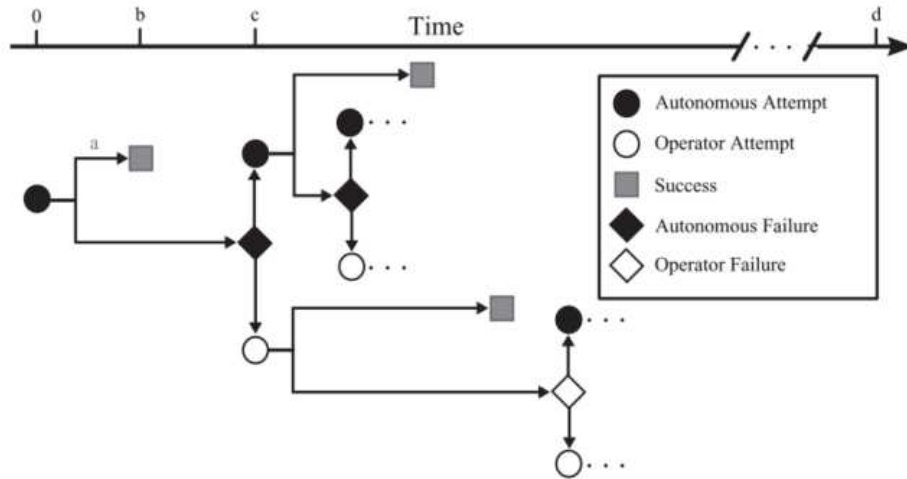


FIGURE 1.13 – Exemple d'arbre de décision (figure tirée de [Sel+06a])

Lyons *et al.* ont proposé dans [Lyo+12] un algorithme permettant de vérifier si une combinaison d'algorithmes, de modèles de robots, de capteurs plongée dans un environnement donné permet de satisfaire un critère de performance (e.g. durée, énergie). Ils vérifient alors probabilistiquement si la mission est réalisable. Dans le cas contraire, les causes potentielles de l'échec estimé sont fournies à l'opérateur afin de l'aider à améliorer itérativement la configuration du robot pour garantir la réalisation de la mission. La vérification est formelle et la mission considérée reste simple. De plus rien n'est proposé pour surmonter des problèmes imprévus lors de l'exécution de la mission.

1.3.2.2 Méthodes non probabilistes

Hernando *et al.* abordent dans [Her+10] la problématique de management (*self-optimising*) de la bande passante entre le robot et le superviseur afin de décider les ressources du robot à utiliser. La solution est trouvée analytiquement à partir de deux fonctions d'utilité représentant la combinaison des ressources du robot (vidéo profondeur, vidéo rgb, laser et sonar) et l'impact des fréquences d'acquisition. Cependant le problème abordé est mono-critère, et sa solution d'affectation est analytique et facilement trouvable.

La décision multicritère est très largement utilisée, particulièrement en productique. La recherche d'une solution satisfaisante prenant en compte des points de vue distincts, voire conflictuels, repose sur le calcul de solution non dominée (ensemble/front de Pareto [CS11] [Ben14]). Outre sa complexité de calcul, le problème de sélection des solutions les plus satisfaisantes reste difficile.

La décision multi-attribut (critères et alternatives) peut se formaliser de la façon suivante (Figure 1.14) : Soit m le nombre de critères, et n le nombre d'alternatives. Les critères sont sous la forme $C_1, C_2, \dots, C_m$ et les alternatives $A_1, A_2, \dots, A_n$. Les critères sont dotés de poids $w_1, w_2, \dots, w_m$ qui représentent leur importance relative en fonction des autres critères. Pour chaque critère, les solutions alternatives ont des "scores locaux" sans dimension : $a_{11}, a_{12}, \dots, a_{mn}$. Les valeurs x_i sont les scores des alternatives i . Cette matrice est appelée matrice des jugements dans [Sch85].

		x_1	.	.	.	x_n
		A_1	.	.	.	A_n
w_1	C_1	a_{11}	.	.	.	a_{1n}
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
w_m	C_m	a_{m1}	.	.	.	a_{mn}

FIGURE 1.14 – Matrice de comparaison alternatives/critères

Il existe 3 grandes familles multicritère selon la méthode d'agrégation choisie [Ben00] :

- Les méthodes d'agrégation complètes :
 - Agrégation puis comparaison où les scores de satisfaction sont tout d'abord agrégés puis les alternatives rangées en fonction du score obtenu (pire cas, moyenne algébrique / géométrique pondérée (MAUT [PBR93])).

- Comparaison puis agrégation où les alternatives sont comparées 2 à 2 puis les résultats agrégés (AHP [Saa08]/ MACBETH³).
- Les méthodes d'agrégation partielle ou de surclassement qui acceptent l'incomparabilité entre alternatives et la non transitivité des préférences (ELECTRE [Roy68]/ PROMETHE [BM05]).
- Les méthodes d'agrégation locale lorsque le nombre d'alternatives à considérer est très important. On part alors d'une solution initiale (la meilleure possible) et cherche à trouver des solutions voisines de meilleure qualité en faisant appel par exemple à des techniques de programmation linéaire multicritère.

Dans [Ben+05] l'objectif est de pouvoir déterminer qu'elles étaient les affectations de ressources humaines possibles pour atteindre une performance donnée au niveau d'un processus. Un processus est vu comme une succession d'activités exécutées séquentiellement ou de façon concurrente. Plusieurs axes de performance ont été étudiés pour l'entreprise (qualité, coût, délais), aux comportements antinomiques.

Au niveau de chacune des activités, le principe d'estimation de performance est basé sur la modulation d'une performance nominale connue, en fonction des savoirs, savoir-être et savoir-faire individuels et collectifs des ressources humaines pouvant être affectées. Ces calculs peuvent être faits hors-ligne. Des lois de composition adaptées à chacun des axes de performance sont alors utilisées pour estimer la performance attendue d'un processus pour un comportement donné et une affectation spécifiée et évaluée vis-à-vis d'un objectif donné. Suite à cette estimation, des fonction d'utilités [Suk97 ; Gro15 ; Med11] permettent de classer les différentes alternatives au niveau de chaque activité.

La difficulté principale du processus d'affectation est qu'il n'est pas possible d'envisager une exploration exhaustive de l'univers des solutions envisageables car ce problème peut être assimilé à un problème de sac à dos donc la complexité est de type NP-complet.

Un ensemble d'algorithmes de résolution de ce type de problèmes est présenté dans [TB06] comme l'algorithme de [KH82] qui est parmi les algorithmes les plus classiques. L'algorithme de recuit simulé (*Simulated Annealing*) [KGV82] est un algorithme méta-heuristique utilisé généralement pour trouver une **solution optimale approchée** de problèmes ayant une forte combinatoire. [UTF95] propose une généralisation de cet algorithme en utilisant une combinaison convexe des différents critères pour avoir une évaluation des solutions. Les solutions de chaque itération sont stockées comme un potentiel optimum Pareto. L'ensemble de solutions est agrégé par la suite pour obtenir une approximation de la solution.

3. <http://www.m-macbeth.com/fr/>

Pisinger [Pis95] propose un algorithme permettant de trouver la **solution optimale** en un temps borné en fonction de la complexité du problème. L'algorithme consiste en une décomposition du problème en n parties qui seront fusionnées deux par deux. La fusion est strictement locale mais les limites supérieures et inférieures sont déduites du dénombrement des autres ensembles.

Bennour *et al.* proposent dans [Ben+05] un algorithme d'une meilleure efficacité en termes de nombre d'itérations nécessaires pour trouver une solution, que pour les algorithmes génétique de Pisinger [Pis95], *Hybrid Taboo-Descent* [HGH99] ou *Simulated Annealing algorithm* [Met+53]. Une solution s'appuyant d'une part sur un classement local, au niveau de chacune des activités, des performances d'affectation, et d'autre part sur un algorithme dichotomique local de complexité ($O(\ln(n))$) pour évaluer la performance du processus complet permet de dépasser le problème d'explosion combinatoire pour trouver très rapidement un ensemble ordonné de solutions d'affectation satisfaisant l'objectif fixé. Une bonne solution est alors trouvée. Elle n'est pas forcément la meilleure mais elle est tout de même proche de cette dernière. Cependant une limitation de cette approche est que les lois de compositions des performances locales (e.g. addition) doivent préserver la relation d'ordre locale. Ainsi trouver une bonne solution à partir d'un ensemble initial de solutions possibles, implique la détermination du sous ensemble composé de tous les solutions possibles (Figure 1.15).

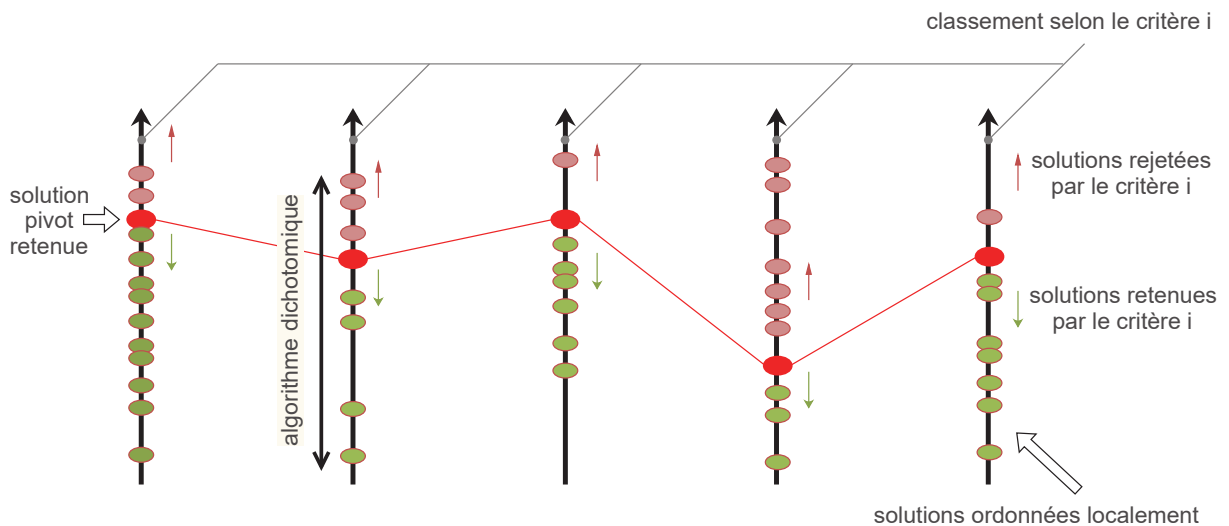


FIGURE 1.15 – Illustration des hypothèses de classement de l'algorithme proposé dans [Ben+05]

Lorsque plusieurs objectifs sont simultanément visés (multicritère), l'utilisation itérative de cet algorithme permet de converger vers un ensemble de solutions couvrant tous les objectifs imposés. La rapidité de résolution permet d'envisager une utilisation en ligne

de l'algorithme d'affectation dès lors que les classements locaux ont été réalisés. Notons que l'ordre de prise en compte des activités et des objectifs a un impact sur la ou l'ensemble de solutions calculées. Une phase de backtracking est également proposée pour améliorer l'efficacité de l'algorithme. Elle permet d'inverser le mécanisme de recherche sur l'ensemble d'activités parcourues afin de trouver une meilleure solution plus proche de la solution optimale. Enfin, cet algorithme a l'avantage d'être simple à mettre en œuvre, efficace et rapide (important en temps réel) et il garantit de trouver une solution si elle existe.

1.4 Positionnement

L'objectif de cette thèse est la proposition d'une approche permettant la gestion dynamique des ressources matérielles et logicielles en mission robotique autonome sous des contraintes de performance. L'analyse du concept d'autonomie a démontré la corrélation entre l'autonomie, la prise de décision et la performance. L'autonomie induit la capacité de faire un choix à un moment donné, la performance sera le critère qui guidera le choix. Cette dernière évalue l'efficacité du système étudié et permet ainsi une adaptation de ce dernier si une dégradation de performance est observée. Nous nous sommes intéressés plus particulièrement à l'**autonomie de comportement** qui se traduit au travers du choix des ressources matérielles et logicielles dans le cadre de notre problématique. Elle consiste à doter le système robotique de la capacité de **décider** quels capteurs, actionneurs et algorithmes sont à mettre en œuvre tout au long de sa mission.

Le concept de performance a été ensuite étudié. En robotique mobile et contrairement à la robotique manufacturière, ce concept n'est pas bien défini dans la littérature même s'il est souvent utilisé. Nous avons noté un certain manque de rigueur et de formalisation de la performance ainsi qu'une myriade de différentes métriques. De façon à palier à ces limites, nous proposerons dans le chapitre suivant un cadre de formalisation de la performance pour les missions robotiques. Cette formalisation s'appuiera sur une définition des **axes de performance** communs à toutes missions robotiques ainsi que la proposition d'une terminologie inspirée de la productique.

Les contraintes imposées par l'environnement, la réglementation ou par l'utilisateur sont les facteurs permettant de piloter le processus de décision. La décision conduisant à l'adaptation dynamique de la configuration du système robotique est déclenchée aussi par l'occurrence de fautes sur ses ressources et les perturbations internes ou externes. Comme l'a montré notre analyse, le processus de décision pour l'affectation des ressources est très souvent formalisé sous la forme d'un problème "sac à dos" multicritère. Les critères

désigneront les **performances contraintes**. L'objectif sera alors de déterminer les configurations matérielles et logicielles à adopter, tout au long de la mission, pour respecter ces critères.

Le chapitre suivant présente une formalisation du concept de la performance dans le cadre de la robotique mobile. Il propose une méthodologie de gestion des ressources d'un système robotique dans le cadre d'une mission autonome sous des contraintes de performance.

Chapitre 2

Proposition d'une méthodologie de gestion de la performance des missions en robotique mobile

Sommaire

2.1	Axes de performance	49
2.1.1	Classification des axes de performance	50
2.1.2	Caractéristiques de la performance	55
2.1.3	Marge de performance	58
2.2	Approche de gestion des ressources matérielles et logicielles proposée	60
2.2.1	Définitions	61
2.2.2	Phase préliminaire (P1)	72
2.2.3	Estimation hors ligne de la performance (P2)	76
2.2.4	Estimation et gestion en ligne de la performance (P3)	80
2.3	Conclusion approche de gestion des ressources	81

2.1 Axes de performance

L'état de l'art qui vient d'être balayé a montré que la définition du concept de performance manque de rigueur et de standardisation en robotique mobile. Cependant beaucoup de critères de performance ont été développés pour comparer l'efficacité d'algorithmes utilisés en robotique (suivi chemin, suivi de cible, planification, etc.) ou différentes approches de conception (robot à roues, robot à pattes, etc.) dans le but de choisir les plus perfor-

manentes. Cette comparaison est réalisée dans des conditions expérimentales reproductibles en simulation ou sur des benchmarks spécifiques. Il faut remarquer que l'ensemble des métriques proposées s'inscrit plutôt dans une logique de mesure de la performance plutôt que d'évaluation de celle-ci. Par ailleurs ces études restent très centrées sur des besoins spécifiques à l'utilisation qui ne sauraient être pertinents dans d'autres contextes de robotique mobile.

Si l'on se projette dans le cadre de cette thèse où l'on s'intéresse à la performance d'une mission en robotique mobile, nous ne retiendrons que certains des points de vue qui ont été évoqués dans l'état de l'art tout en proposant une définition et une classification des différents axes de performances, de leur typologie, ainsi que du concept de marge de performance.

2.1.1 Classification des axes de performance

Nous distinguons en premier lieu les **performances réalisées par le système robotique** $Perf_{SR}$ des **performances contraintes** $Perf_C$ imposées pour la mission par l'utilisateur ou par des règlements. Bien évidemment, selon leur type (continue/booléenne), l'objectif est de les comparer ($Perf_{SR}$ et $Perf_C$) via une relation d'ordre (performance continue) ou par comparaison logique (vrai/faux) :

- **Continue** : C'est le cas des performances qui évoluent de façon cumulative, continue et uniforme telles que l'énergie, la stabilité et la durée.
- **Booléenne** : Elle représente les performances qui peuvent être vraies ou fausses sans présenter d'état intermédiaire. C'est le cas de la sécurité et la localisation.

De nombreux critères de performance ont été proposés en robotique mobile. Ils sont généralement utilisés pour évaluer un système robotique ou le comparer avec d'autres systèmes. Cependant certaines facettes de performance sont très souvent implicitement traitées sans être explicitement évaluées (**Sécurité** et **Localisation**). Même s'il est central pour la réalisation de la mission, l'**Énergie** est le plus souvent éludée. Enfin, la **Stabilité** qui est indispensable pour assurer la commande du système robotique n'est pas considérée comme un critère de performance mais est abordée que sous le point de vue purement automatique. Sans prise en compte de ces performances, aucun système robotique mobile ne serait capable d'exécuter sa mission pour lequel il a été conçu. C'est pourquoi ils seront désignés comme **axes principaux de performance** (Figure 2.3).

Évidemment l'utilisateur du système robotique cherche à évaluer le(s) service(s) rendu(s). Ces besoins utilisateurs constituent **des axes secondaires** spécifique(s) au contexte (**Durée** de la mission, trouver une cible, accompagner un visiteur, etc). Nous allons

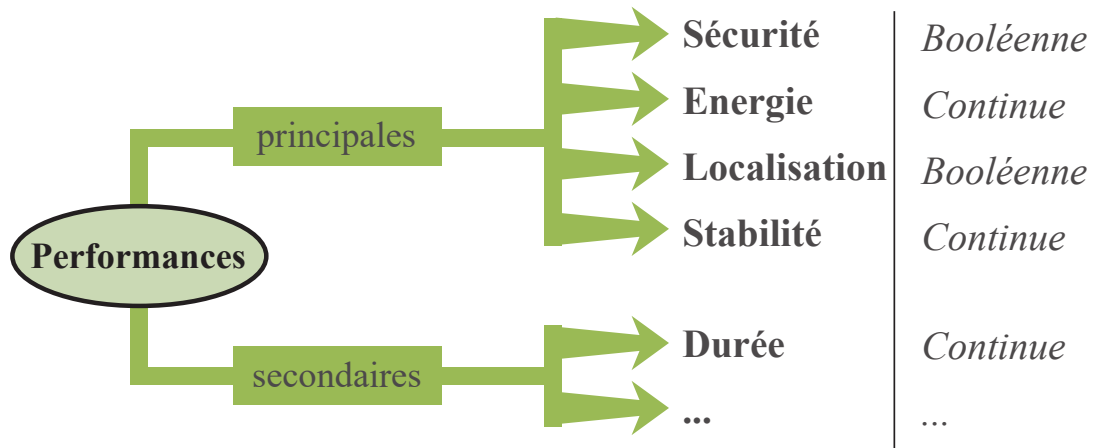


FIGURE 2.1 – Classification des axes de performance

détailler dans la suite les 4 axes de performance principaux et la durée comme axe de performance secondaire particulier.

2.1.1.1 Axes de performance principaux

Dans le cadre de la robotique mobile, les principaux axes de performance sont ceux indispensables pour la réalisation de la mission quelque soit le robot, la mission et l'environnement considérés. Nous proposons de retenir 4 axes principaux de performance : La sécurité, l'énergie, la localisation et la stabilité.

a. Sécurité

L'aspect sécuritaire est primordial et est indéniablement le plus prioritaire des 4 axes de performance proposés. Il est impossible de l'écarter et il est indissociable des applications en robotique mobile. Il est implicitement ou explicitement considéré dans la plupart des articles analysés puisque la sécurité des biens et des personnes est obligatoire [CVO10] [BHP08]. Le système robotique ne doit pas présenter de danger pour lui même ou pour son environnement.

Lié à l'interaction que le robot peut avoir avec son environnement, deux points de vue peuvent être considérés en fonction de la source du danger :

- **De l'environnement sur le robot** : cela désigne toutes agression de l'environnement envers le robot amenant à une dégradation matérielle qui peut perturber ou suspendre son fonctionnement telles que fuite, radiation, corrosion, surchauffe interne, choc, etc. Par exemple, assurer et vérifier l'étanchéité est impératif pour un

robot sous-marin, tout comme ne pas dépasser la profondeur limite pour laquelle il a été conçu. Il peut aussi être étendu à la capacité de ne pas chuter [TC 12] (dans des escaliers par exemple) où à la capacité de détecter des obstacles de différente nature (objets opaques (laser ou sonars (US)), objets transparents (US)).

- **Du robot sur l'environnement** : l'environnement représente tout qui est extérieur au robot (homme, nature, bâtiment, autres robots, etc.). Le robot interagit nécessairement avec (ou non) / dans / sur beaucoup d'éléments de son environnement. Son action doit être positive comme assister, guider un être humain ou neutre lorsque l'on cherche à éviter les interactions physiques avec le robot. Tout effet négatif peut nuire ou entacher le service rendu par le robot comme une pollution de la nature, étincelle dans un environnement gazeux, choc avec l'homme ou autres robots, etc.

La prise en compte de cet axe de performance réside dans l'évaluation de la conformité sécuritaire de la conception matérielle du système robotique (aspect statique) mais aussi l'évaluation des performances des différents algorithmes assurant que le service rendu par le robot n'agresse pas involontairement son environnement (aspect dynamique).

b. Énergie

L'axe énergétique semble tout aussi incontournable pour les robots mobiles autonomes qui, contrairement à leurs homologues industriels, ne peuvent être directement reliés à une source d'énergie. Cette dimension est le plus souvent négligée dans la littérature qui considèrent que le robot dispose de l'énergie nécessaire pour mener à bien sa mission. Cette dimension est évidemment plus prégnante pour des mission robotiques extraterrestres [Mic+02] [Pre03] [Tun07].

Le robot mobile doit donc embarquer (ou régénérer) son énergie de façon à pouvoir assurer l'alimentation de ses capteurs et son actionnement. Sans oublier l'alimentation, souvent indépendante, des capacités de calcul embarquées sans lesquelles il ne peut assurer sa mission. Cette dernière doit donc pouvoir être assurée uniquement à l'aide de l'énergie embarquée (batteries), où une tâche de rechargement doit pouvoir être planifiée par le robot si une station de recharge est disponible dans l'environnement.

Quoi qu'il en soit l'évolution autonome dans le cadre d'une mission longue impose que le robot soit capable d'estimer avant son départ, s'il dispose d'assez d'énergie pour la mener à bien. Par conséquent, une estimation de l'énergie consommée par les différentes fonctionnalités, basé sur la description a priori de la mission est nécessaire. Il doit être aussi capable de décider en cours de mission si, en raison d'aléas de fonctionnement (trajet

plus long, etc.), s'il dispose encore d'assez d'énergie pour achever son travail. Dans le cas contraire il devra changer de configuration matérielle et logicielle ou changer d'objectifs (chemin plus court, se recharger, etc.).

Outre l'importance de la dimension énergétique pour la réalisation de la mission, il faut assurer la réalisation des différentes fonctionnalités requises par l'axe sécuritaire (perception, réaction à un obstacle) mais aussi, évidemment, à la capacité de localisation du robot.

c. Localisation

Cette dimension de la performance a largement été abordée dans la littérature. Elle est indissociable de l'existence même d'un robot mobile. Lors de son déplacement, le robot doit être capable de se localiser dans son environnement. Dans les missions d'exploration, le robot n'est pas en mesure de se localiser au début de sa mission mais il doit le faire plus tard pour finir sa mission et retourner à la base (e.g. SLAM).

Le robot mobile doit être à même de se localiser (plutôt en absolu) au sein d'une carte de l'environnement qu'elle soit préalablement connue, ou en construction (exploration). Cette dimension de la performance peut aussi se rapporter à la capacité de manipulation du robot qui doit être à même de localiser ses organes de préemption avec précision et relativement à l'objet à manipuler de façon à pouvoir agir avec son environnement. Dans les deux cas la précision de la localisation est importante et celle-ci peut évoluer dynamiquement, en fonction de l'environnement et à fréquence variable en fonction de la présence d'amers et des performances de l'algorithme de localisation. La précision de localisation dépend de la nature de l'environnement, des capteurs utilisés et de la fréquences de rafraichissement de ses données (voulue par l'utilisateur ou imposée par le matériel).

Une forte corrélation entre l'axe de localisation et les précédents points de vues de la performance existe. En effet, la localisation indique, entre autres, la position relative entre le robot et les éléments de son environnement ce qui impacte directement la dimension sécuritaire de la mission. D'autre part, la connaissance de la position courante du robot est une donnée indispensable pour estimer l'énergie nécessaire pour finir la mission.

d. Stabilité

C'est le critère d'évaluation de la mobilité des systèmes de locomotion. Ils sont en mesure de faire des franchissements et des évitements d'obstacles. La commande du robot mobile doit permettre d'assurer la stabilité de la boucle du contrôle quelque soit le contexte

de la mission. C’est une condition capitale pour assurer la mobilité attendue. La stabilité est abordée de point de vue de la navigation (mouvement : automatique) et décrochage (risque de renversement de la structure : mécanique).

Pratiquement, la stabilité dépend du matériel, logiciel et des contraintes architecturales (e.g. fréquence CPU). Par ailleurs, la stabilité est aussi fortement conditionnée par l’interaction entre le robot et l’environnement (glissement, patinage, courant d’air ou d’eau). Cela fait de cet axe de performance un challenge rarement traité par ses différentes facettes et le postulat utilisé est que si la boucle de contrôle est assez rapide alors la stabilité des algorithmes de contrôle sera assurée.

La stabilité d’un robot impacte directement sa sécurité et son énergie. La contrôlabilité est essentielle pour pouvoir agir et réagir face aux potentiels dangers. De plus, un robot non stable dérive de sa trajectoire et parcourt plus de distance que prévue. Cela fausse donc la prévision d’énergie consommée.

2.1.1.2 Axes de performance secondaires

Si les axes précédents sont indispensables pour assurer l’exécution de toute mission robotique mobile, ils sont en général complétés par d’autres dimensions permettant d’estimer la qualité du service rendu à l’utilisateur.

Tout service attendu du robot représente un axe de performance de ce dernier. C’est un travail caractérisant la mission et qui ne relève pas des fonctionnalités de base du robot (se déplacer, se localiser, etc). La qualité de ce travail doit être évaluable avec des métriques précises afin de décider du degré de complétude de la mission. Ce point de vue de la performance figure généralement dans la littérature comme un constat. Les métriques proposées sont utilisées pour évaluer/comparer les performances de systèmes robotiques ou d’algorithmes, capteurs ou actionneurs au sein du même système. Les indicateurs de performance peuvent aussi bien être la précision d’une carte construite [BHP08], le nombre de points d’intérêt exploré, la surface nettoyée [Pal+10], etc. Ces performances dépendent des capteurs/actionneurs pouvant être utilisés, des algorithmes embarqués et des capacités de calcul disponibles ainsi que de l’environnement dans lequel est exécutée la tâche robotique ou la mission.

L’axe durée reste une caractéristique centrale de la réalisation d’une mission qui doit être achevée en temps fini. Il représente un axe de performance secondaire qui a la particularité d’être, assez souvent, imposé par les donneurs d’ordre pour l’exécution d’une mission sans qu’il soit, généralement, le but principal de la mission comme dans [BCP95a] et [TKI11a] où le but principal est la gestion de l’énergie sous une contrainte de durée de mission.

Le temps nécessaire au robot pour réaliser sa mission ne correspond pas nécessairement à des tâches où le robot effectue un déplacement (de sa base ou d'un organe de manipulation). En effet le robot peut aussi s'arrêter par exemple pour attendre un ordre, décider ou changer ses objectifs, attendre une transition, acquérir ou envoyer des données ou attendre la fin d'un traitement informatique. Notons cependant que l'arrêt cinématique n'est possible que pour les robots terrestres. Cette caractéristique n'est pas partagée par les drones aériens ou les robots sous-marins.

Si nous voulons piloter les performances des robots du point de vue service rendu, nous ne pouvons pas nous limiter à mesurer certaines métriques, il faut aussi développer des lois d'estimation des ces performances pour pouvoir prédire leur évolution tout au long de la mission.

2.1.2 Caractéristiques de la performance

2.1.2.1 Notions liées à un axe de performance

Certaines notions liées à la performance ont besoin d'être définies. Pour ce faire nous les avons extraites du domaine de l'entreprise où ce concept est fréquemment utilisé [MBC01] [Tah03]. Ces définitions peuvent être projetées sur notre domaine d'intérêt. Ainsi, pour chaque axe de performance on associe (Figure 2.2) :

- **Unité** : Grandeur qui permet de mesurer des quantités qui de nature en fonction d'une valeur unitaire (seconde, mètre, Newton, etc).
- **Mesure** : Détermination d'une grandeur par comparaison avec une échelle.
- **Indicateur** : Donnée quantifiée qui mesure l'efficacité de tout ou partie du système robotique (énergie consommée, précision de la localisation). La performance réalisée par le système robotique $Perf_{SR, axe}^{ind_i}$ selon l'axe x peut être mesurée par l'indicateur de performance ind_i . Par exemple, pour un axe de performance désignant la qualité d'un suivi de chemin, l'écart latéral et angulaire peuvent être des considérés comme des indicateurs de performance.
- **Référent** : Cible vers laquelle on se rapporte pour évaluer une grandeur mesurée. C'est une contrainte qui peut être définie par l'utilisateur, loi, etc. En d'autres termes, c'est une valeur maximale, minimale ou à atteindre qui permet de qualifier la performance mesurée. Le référent ref_i selon (relative à l'indicateur ind_i) l'axe de performance x est dénoté par la performance contrainte $ref_i = Perf_{C, axe}^{ind_i}$.
- **Estimation** : Association d'une plage de confiance à une mesure.

- **Évaluation** : Interprétation de la mesure d'un indicateur de performance par rapport à un référent (supérieur, inférieur ou égal).
- **Inducteur** : Levier sur lequel on intervient pour influencer la performance d'un système. Par exemple pour piloter la durée d'un trajet, on peut agir sur la distance à parcourir et la vitesse moyenne de déplacement. Les inducteurs de performance permettent de piloter la performance d'un système.

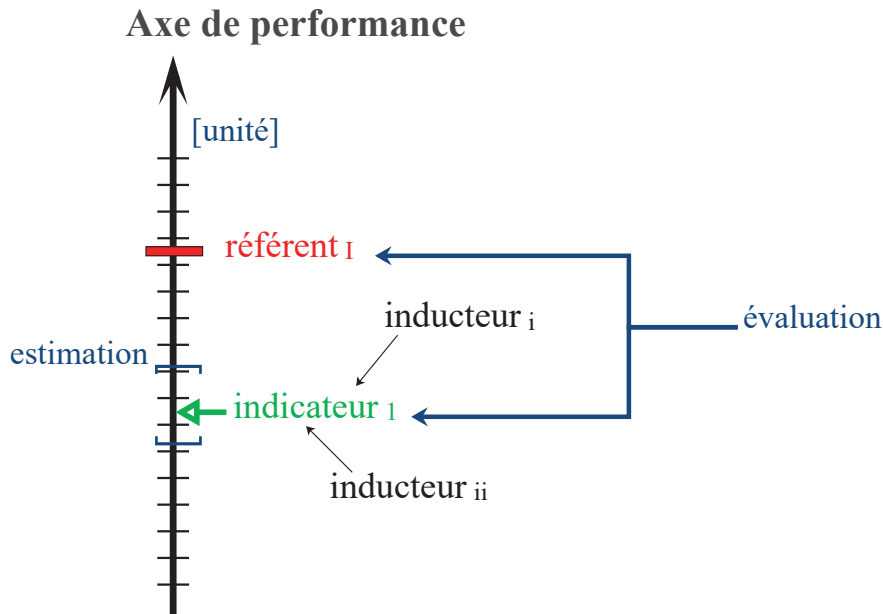


FIGURE 2.2 – Notions liées à la performance

Nous pouvons associer à un seul axe de performance plusieurs référents et indicateurs évalués deux à deux ou plusieurs indicateurs évalués par rapport à un seul référent. De plus, les inducteurs de performance peuvent agir sur plusieurs indicateurs associés à un ou plusieurs axes.

2.1.2.2 Classification des performances

Dans le cadre d'une mission robotique, différentes classes de performances $Perf_{SR}$ peuvent être identifiées pour un système robotique (Figure 2.3) :

- La **performance estimée** prédite à partir d'une formulation (modélisation) pré-établie, pour un indicateur donné. Cette formulation, pour un scénario définissant le déroulement prévu d'une partie ou de la totalité d'une mission, permet de calculer la performance hors ligne (performance a priori), puis en ligne en cas de remise en cause, ou d'ajustement, la performance liée à l'exécution d'une mission.

- La **performance instantanée** est mesurée pour un indicateur de performance donné, à un instant t d'une mission. Elle peut résulter d'une observation et constituer une donnée expérimentale empirique. Par exemple un temps écoulé ou une énergie consommée. Elle peut aussi être le résultat d'un calcul issu d'une formulation telle que la surface couverte par un robot de nettoyage (calculée en fonction de la taille du robot et de la distance parcourue). Dans ce cas l'écart entre le fait expérimental et le modèle permet de définir un écart ou une dérive. La prise en compte de cette dérive permet d'ajuster la performance estimée.
- La **performance finale** qui mesure la performance selon un indicateur donné, au terme de la mission.

Ces notions de performance estimée, instantanée et finale sont évoquées également dans un contexte de robotique mobile dans la thèse de Lampe [LT06].

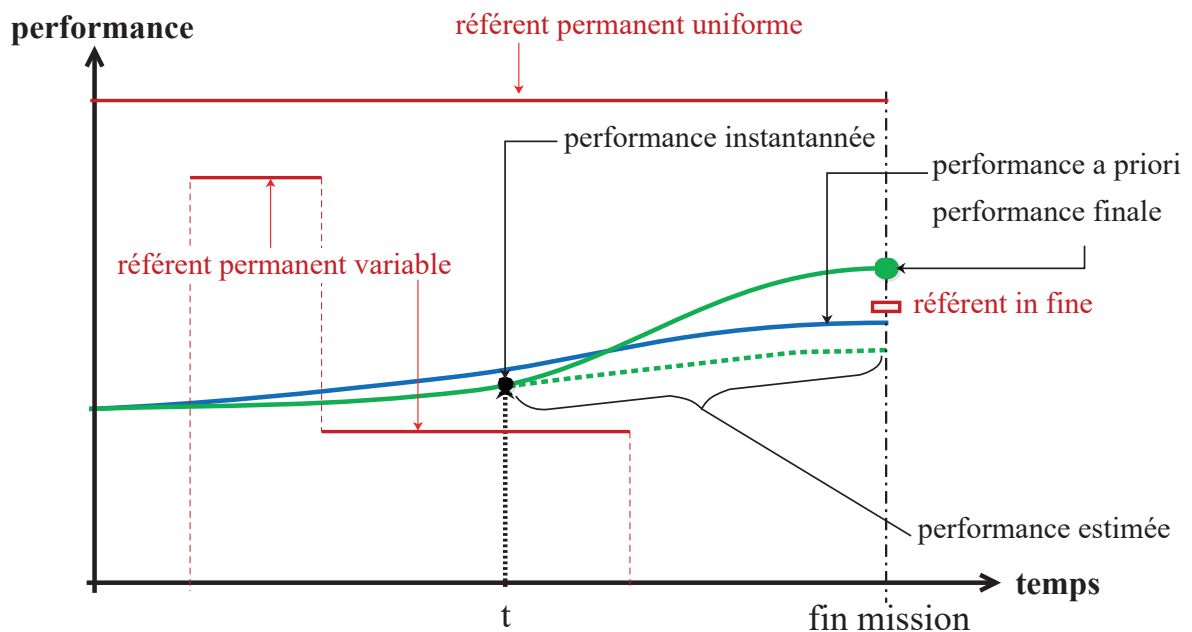


FIGURE 2.3 – Les classes de performance du système robotique $Perf_{SR}$ et performance contrainte $Perf_C$

Les contraintes de performance $Perf_C$ permettent de traduire les objectifs de la mission en un ensemble de critères qualitatifs (bon, mauvais, acceptable) et ou quantitatifs (valeur discrète, intervalle à domaine continu, discret). On peut les classifier comme suit :

- Des contraintes de performance permanente définie sur un ou des intervalle(s) continu(s). Certaines devant toujours être satisfaites (permanente) tout au long de la mission comme la sécurité et la stabilité. D'autres ne doivent être vérifiées que dans certaines zones ou intervalle de temps (localisation). Les contraintes de performance

imposées peuvent être **uniformes** (constante) tout au long de la mission ou **variables** (dynamiques) en fonction de la position spatio-temporelle du robot ou de la tâche en cours d'exécution. Par exemple un robot doit se déplacer avec plus de précision lorsqu'il doit se connecter à une station de recharge que lorsqu'il se déplace dans un espace ouvert.

- Les contraintes de performance (**réfèrent *in fine***) doivent être satisfaites en fin de mission (énergie, durée).

2.1.3 Marge de performance

La marge de performance de l'indicateur i selon un *axe* de performance est notée $M_{axe}^{ind_i}$. Elle représente l'écart entre la contrainte de performance exigée (réfèrent permanent ou *in fine*) et la performance estimée ou réalisée (fin de mission) par le système robotique.

Évidemment les modèles d'estimation de performance ne sont qu'une évaluation que l'on souhaite la plus fidèle possible à la réalité. En effet ces mécanismes n'intègrent que de façon imparfaite les conditions expérimentales réelles. Par ailleurs, nous ne pouvons prendre en compte directement toutes les perturbations auxquelles va nécessairement être confrontée la mission (pannes, obstacles), que ce soit lors de l'occurrence de fautes internes engendrant une dégradation des fonctionnalités du robot, où induites par des événements externes engendrant une déviation locale du comportement attendu lors de la description de la mission. Cela va induire des dérives de performance par rapport au scénario prévu. Ainsi, ces dérives observées doivent être identifiées en comparant la performance instantanée du robot vis-à-vis de celle attendue, de façon à pouvoir recalculer et actualiser les mécanismes de prédiction (estimation) permettant d'assurer que les performances attendues en fin de mission pourront bien être obtenues. La marge de performance estimée est alors actualisée tout au long de la mission pour conclure sur la réalisabilité de cette dernière selon le point de vue de performance considéré (Figure 2.4).

Cependant, les conséquences d'une dérive de performance, souvent pénalisantes pour l'utilisateur, ne sont pas nécessairement catastrophiques pour l'exécution d'une mission. En effet les marges de performance pourront permettre de supporter les dégradations de performances induites par les perturbations inhérentes (erreur de modèles d'estimation induisant une mauvaise estimation) au déploiement de missions robotiques dans un milieu ouvert et dynamique, sans pour autant remettre en cause les objectifs de performance imposés par l'utilisateur. Nous pouvons assimiler les marges de performance à une "réserve de performance".

A notre connaissance, cette notion de marge de performance bien que naturelle, n'est

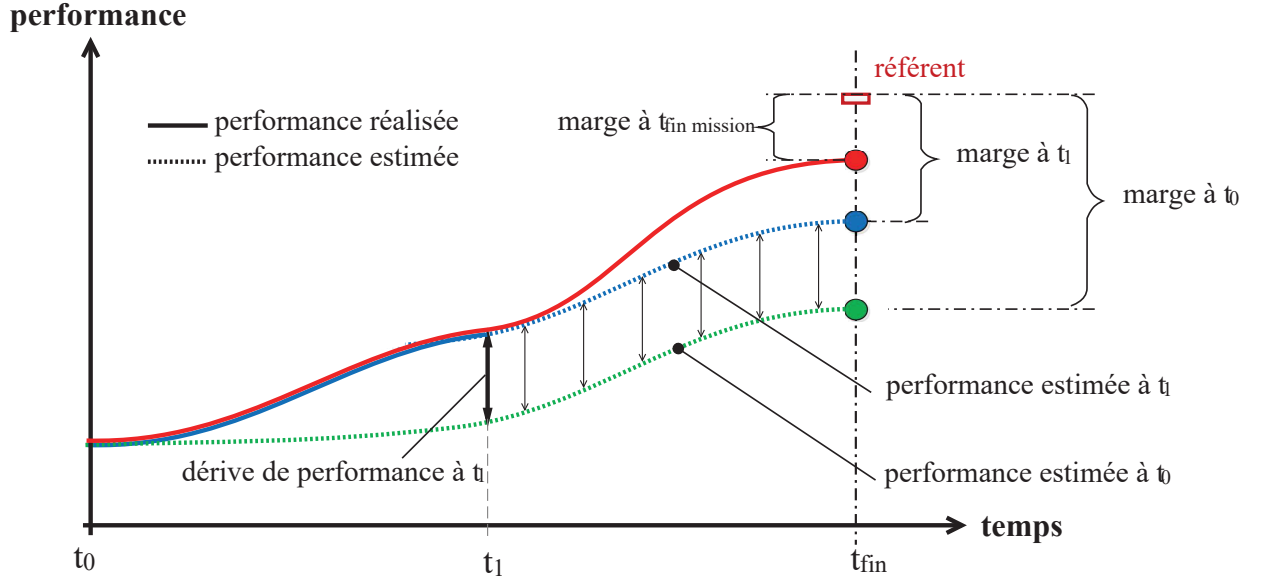


FIGURE 2.4 – Notion de marge de performance

pas fréquemment utilisée, dans le cadre scientifique qui nous intéresse. En automatique linéaire, en revanche, elle mesure le degré de stabilité d'un système en boucle fermé et permet de conclure quant-à sa stabilité (marge positive), son instabilité (marge négative) ou sa limite de stabilité (marge nulle) en évaluant sa distance vis-à-vis d'un point critique [LT06].

Nous considérerons, pour un axe de performance donné, que le point critique $Perf_{C, axe}^{ind_i}$ d'un indicateur de performance ind_i correspond au niveau de performance minimal (resp. maximal) devant être (resp. ne pas être) dépassée par le robot pour que la mission puisse être réalisée (vis-à-vis de cet indicateur de performance selon l'axe de performance considéré). Soit $\widehat{Perf}_{SR, axe}^{ind_i}$ la performance estimée. La marge de performance est alors calculée (en fonction de la nature des axes de performance considérés) comme suit :

Performance minimale à assurer :

$$M_{axe}^{ind_i} = \widehat{Perf}_{SR, axe}^{ind_i} - Perf_{C, axe}^{ind_i} \quad (2.1)$$

ou

Performance maximale à assurer :

$$M_{axe}^{ind_i} = Perf_{C, axe}^{ind_i} - \widehat{Perf}_{SR, axe}^{ind_i} \quad (2.2)$$

La valeur de la marge de performance a toujours la même interprétation :

- Si $M_{axe}^{ind_i} > 0$ alors la mission peut être réalisée vis-à-vis de l'indicateur ind_i suivant l'axe de performance.

- Si $M_{axe}^{ind_i} < 0$ alors la mission n'est plus réalisable vis-à-vis de l'indicateur ind_i suivant l'axe de performance.
- Si $M_{axe}^{ind_i} = 0$ alors la mission est réalisable mais sans possibilité de gestion des aléas.

Nous avons proposé dans cette partie une formalisation de la performance dans le contexte de la robotique mobile. Une première classification a été faite pour différencier entre des axes de performance principales (sécurité, énergie, localisation et stabilité) qui sont "vitaux" pour tout système robotique mobiles et des axes de performance secondaires qui du point de vue utilisateur sont essentiels pour la réalisabilité de la mission robotique mais qui ne sont pas nécessaire pour la "survie" du robot. Certaines notions liées au concept de performance ont été présentées. Elles permettent de la caractériser et d'interpréter ces résultats. Finalement, le concept de marge de performance a été défini. C'est lui qui permettra de juger de la réalisabilité de la mission en indiquant le degré de tolérance aux potentielles dérives.

Le concept de performance étant maintenant clairement défini, la démarche de gestion des ressources matérielles et logicielles d'un système robotique mobile sous des contraintes de performance va maintenant pouvoir être présentée.

2.2 Approche de gestion des ressources matérielles et logicielles proposée

Cette partie propose une approche de gestion des ressources matérielles et logicielles d'un système robotique mobile guidée par des contraintes de performance. Tout d'abord il faut spécifier les inducteurs impactant la performance, donner une typologie de ces derniers afin d'identifier ceux qui sont pilotables avant et en cours de la mission ainsi que ceux qui ne le sont pas. Une formalisation générique d'une mission robotique, de notre point de vue, se montre indispensable. Cette formulation permettra de projeter les contraintes de performance tout en montrant leur impact sur les ressources utilisables ainsi que sur leurs paramètres.

Trois phases composent la démarche proposée. Une phase préliminaire consiste à identifier les indicateurs de performance et leurs interdépendances pour modéliser la performance vis à vis d'un (des) indicateur(s) suivant un axe considéré. Une estimation hors ligne de la performance globale du système robotique sur la mission traitée est entamée. Une affectation de ressources doit être trouvée tout au long de la mission. Si une solution existe, la phase d'évaluation et de gestion en ligne des ressources est poursuivie en parallèle avec l'exécution de la mission. Elle permet de vérifier si, suite aux éventuels aléas (dérives de performance et/ou fautes sur les ressources), les contraintes de performance

sont toujours respectées. En cas d'impossibilité de réalisation de la mission avec les configurations prévues, une autre solution doit être trouvée pour poursuivre la mission (si elle existe).

2.2.1 Définitions

Avant de présenter l'approche de gestion de ressources, nous allons commencer par proposer une formalisation de la notion de mission, des ressources d'un système robotique mobile, les contraintes susceptibles d'impacter le déroulement de la mission, les principaux inducteurs de performance et enfin l'algorithme général de gestion des ressources.

2.2.1.1 Mission et ressources : Définitions

Vue son importance, nous allons commencer par une description générique d'une mission robotique ainsi que les différentes hypothèses considérées (Figure 2.5).

Définition 1:

Une **Mission** robotique peut être assimilée à l'exécution en **séquence** de n_{Obj} **Objectifs** O .

$$Mission = \{Objectif\}_{seq} \quad (2.3)$$

Définition 2:

Un **Objectif** est une action de haut niveau telles que aller à un endroit, trouver un objet, analyser des données, etc. Il est exécuté par une ou plusieurs **Tâches** T **concourante(s)** (n_T^A est le nombre de tâches pour l'activité A). Le domaine d'application d'un objectif peut être défini spatialement (action de mobilité sur une **distance**) ou temporellement (action immobile pendant une **durée**).

$$Objectif = \{Tâche\}_{conc} \quad (2.4)$$

Définition 3:

La **Tâche**, elle même, peut être réalisée avec différentes **Options d'Implémentation de Tâche OIT** (configuration matérielle et logicielle (paramétrée) des ressources du système robotique). Une *OIT* doit être **sélectionnée** pour réaliser une tâche.

$$Tache = \{OIT\}_{sél} \quad (2.5)$$

Définition 4:

Une **Alternative d'Implémentation AI(P)** représente une combinaison d'*OIT* par tâche permettant de réaliser l'objectif. Elle peuvent avoir un ou plusieurs paramètres $P = \{p_0, p_1, \dots\}$.

La réalisation d'un objectif nécessite de choisir une *OIT* pour chacune des tâches associées à l'objectif considéré. Une *AI(P)* correspond à une des combinaisons d'*OIT* envisageables, de paramètres P .

2.2.1.2 Contraintes

La mission robotique se réalise sous différentes contraintes provenant de plusieurs origines.

Définition 5:

Des **contraintes** c_j environnementales (fiabilité capteurs, algorithme), matérielles (limite de vitesse linéaire, angulaire du robot) ou réglementaires (vitesse en milieu peuplé) peuvent s'appliquer sur la mission considérée (Figure 2.7). Elles peuvent être définies spatialement ou temporellement.

Outre les contraintes de performance (principales et secondaire), nous pouvons citer classes de contraintes qui doivent être considérées.

- **Légale** : Ce sont les contraintes provenant de la législation. Elles sont principalement liées à la sécurité. Elles englobent des règles d'utilisation des engins mobiles dans des environnement peuplés (vitesse, sonorité). D'autres règles sont établies vis-à-vis la protection de l'environnement des systèmes robotiques mobiles tel que l'émission des gaz dangereux et de radiation [ISO11b] [ISO11a].
- **Physique et fonctionnelle** : Elles représentent les caractéristiques matérielles (vitesse, accélération, dimensions, poids, capteurs, actionneurs) et logicielles (fré-

quence, mémoire, résolution, unités de calcul) du système robotique mobile. Ces limites sont généralement définies par le constructeur.

- **Environnementale** : Les éléments constituant l'environnement dans lequel le robot évolue peuvent présenter des contraintes particulières liés à l'adéquation entre leurs propriétés (opacité, luminosité, transparence) et les technologies utilisées par le robot. À titre d'exemple, un robot sous-marin ne peut pas voler et un capteur laser n'est pas efficace face à un obstacle transparent (vitre).

La projection de ces contraintes sur la mission engendre, un découpage des objectifs de la mission en zone spatiale/temporelle de contraintes invariantes. Ces zones sont appelées activités (Définition 6). La prise en compte de ces contraintes peut **disqualifier** certains *AI* (mobilisant des ressources non fiables) et/ou **contraindre** des paramètres $P = \{p_0, p_1, \dots\}$ (s'il(s) existe(ent)) comme imposer une vitesse maximale ou une précision minimale d'un capteur.

Définition 6:

La mission se décompose en une **séquence** de n_{Act} **activités** **A** où chacune se réalise sous des contraintes invariantes. Chaque activité peut se réaliser avec une $AI(P)$ parmi l'ensemble de *AI* possibles $\{AI\}$. On appelle cette séquence, le **Scénario** de Mission **SM**.

$$SM = \{A_1^{\{AI(P)\}_1}, A_2^{\{AI(P)\}_2}, \dots, A_x^{\{AI(P)\}_x}, \dots, A_{n_{act}}^{\{AI(P)\}_{n_{act}}}\}_{seq} \quad (2.6)$$

Définition 7:

Les objectifs, activités et contraintes sont définies sur des zones qui sont délimitées spatialement par leurs coordonnées initiales et finales (Figure 2.6). Celle ci représentent, par rapport au début de la mission, les coordonnées linéaires de début et de fin de la zone considérée. Selon la vitesse adoptée, une dualité distance/temps peut être déduite.

Reprenons maintenant avec plus de détails les différentes configurations d'implémentations au niveau d'une activité *A*. Comme on l'a dit précédemment, l'activité est réalisée avec une ou plusieurs tâches. Chaque tâche *T* de l'activité *A* est réalisée avec $n_{OIT}^{T/A}$ options d'implémentation de tâche *OIT* (Figure 2.7). Les *OIT* peuvent être paramétrables (ie. fréquence, vitesse). Ainsi l'alternative d'implémentation *AI* composée par une *OIT* par tâche, hérite les paramètres correspondant à ses *OIT*. Le nombre local d'*AI* par activité n_{AI}^A est la combinatoire des différentes combinaisons possibles d'*OIT* (2.7).

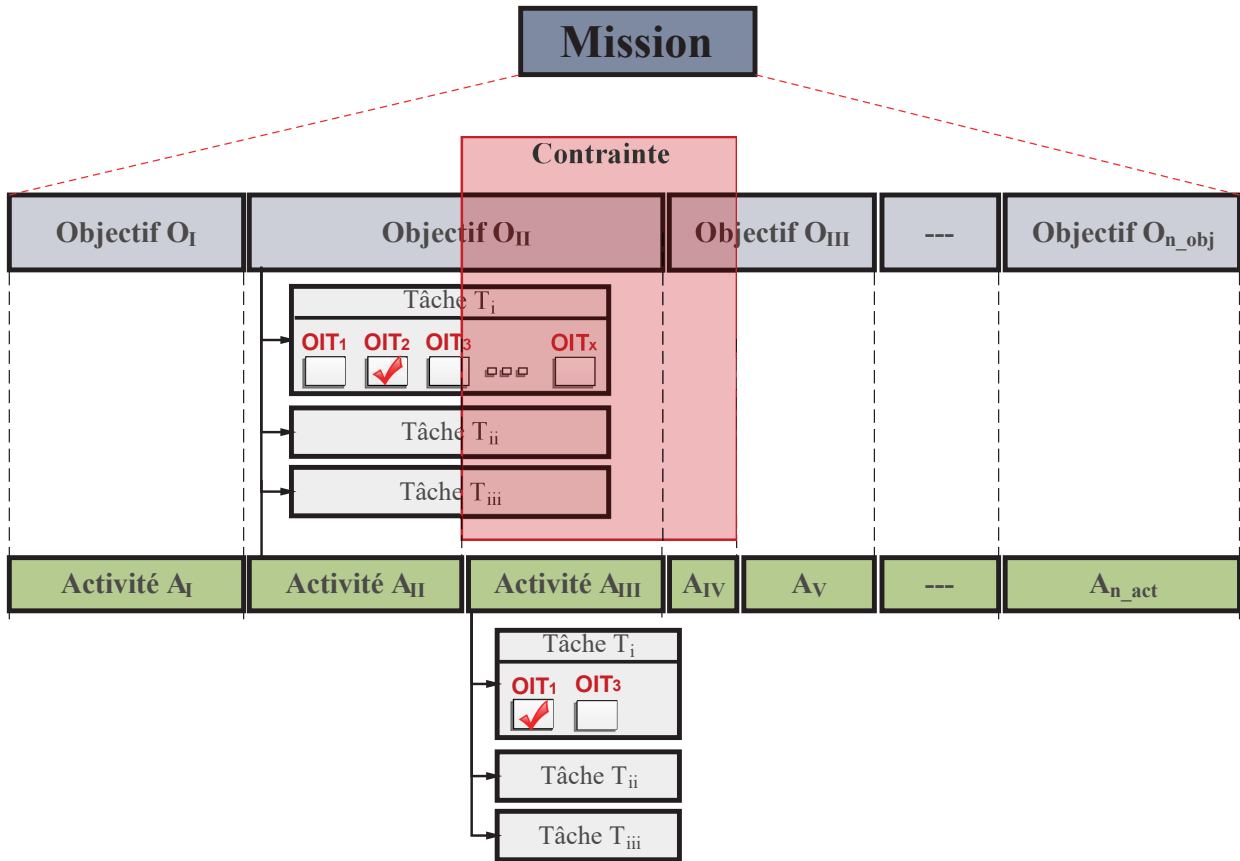


FIGURE 2.5 – Hypothèses de description d'une mission robotique

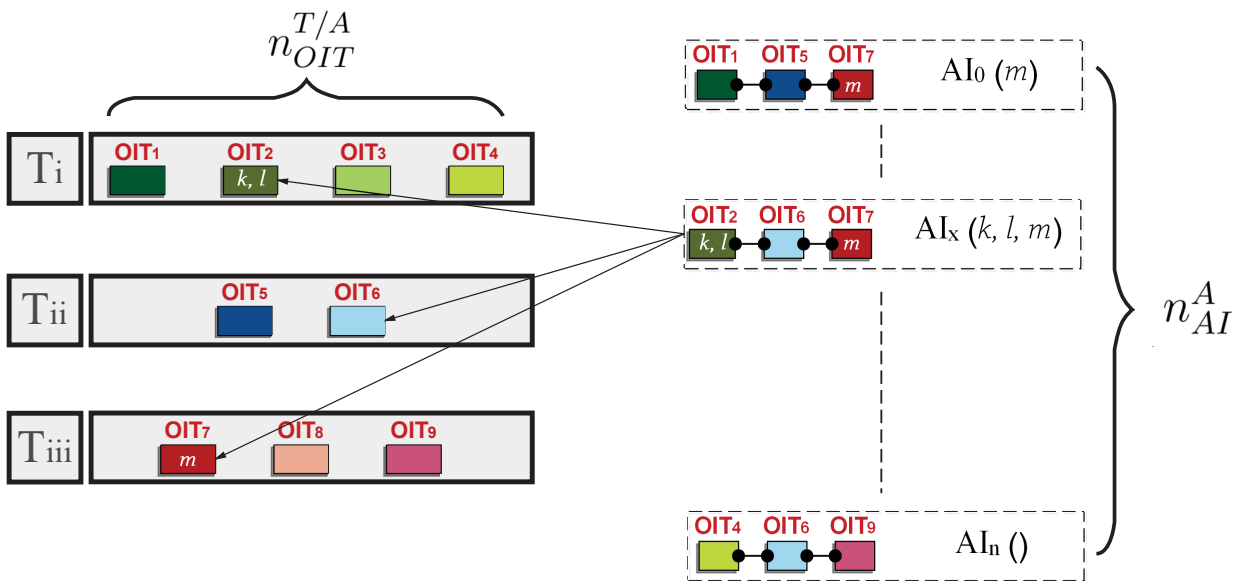


FIGURE 2.7 – Exemple de compositions des alternatives d'implémentations AI.

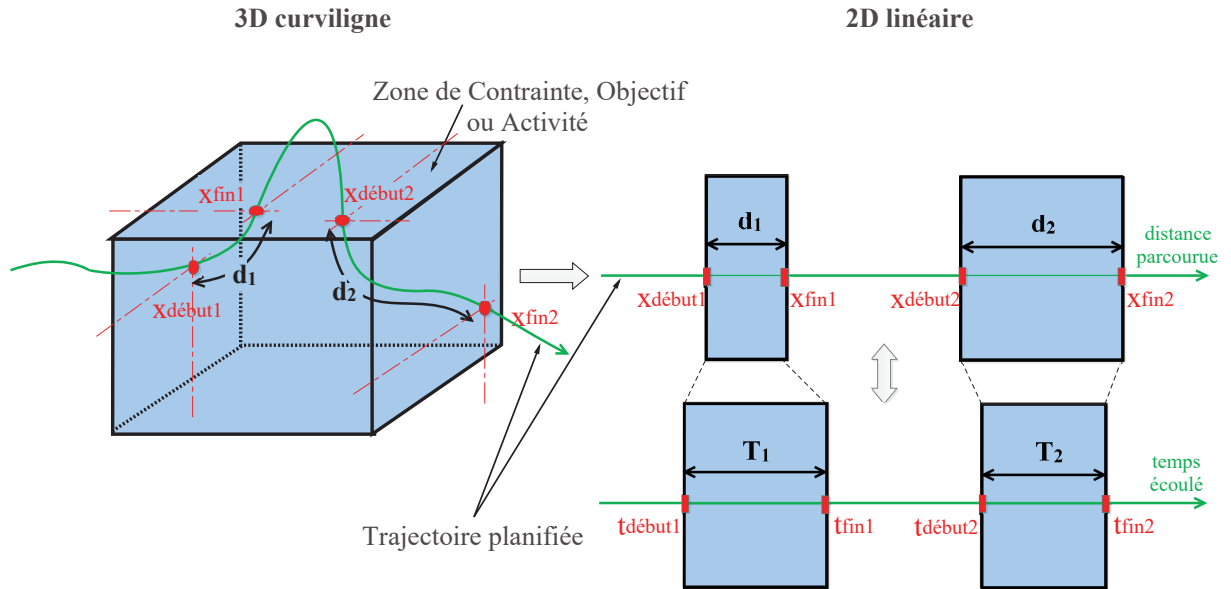


FIGURE 2.6 – Projection unidimensionnelle des objectifs, activités et contraintes.

$$n_{AI}^A = \prod_{i=1}^{i=n_T^A} n_{OIT}^{T_i/A} \quad (2.7)$$

Le tableau 2.1 récapitule les valeurs des différentes variables correspondant à l'exemple de la figure 2.7.

Tableau 2.1 – Tableau récapitulatif des nombres de tâches et OIT.

n_T^A	$n_{OIT}^{T_1/A}$	$n_{OIT}^{T_2/A}$	$n_{OIT}^{T_3/A}$	n_{AI}^A
3	4	2	3	24

2.2.1.3 Complexité

Rappelons que le but est de déterminer pour chaque activité de la mission, une AI permettant de la réaliser. L'ensemble de solutions possibles caractérise la complexité du problème de décision. Elle est définie comme suit :

Définition 8:

La complexité de décision globale du problème correspond à l'espace d'état global à explorer. Elle est égale au Nombre Global de combinaisons possibles d'Alternatives d'implémentation NGA . C'est le produit du nombre d'AI par activité (Figure 2.8).

$$NGA = \prod_{i=1}^{i=n_{act}} n_{AI}^{A_i} \quad (2.8)$$

Comme nous le verrons par la suite, dès que n_{act} et n_{AI} augmentent la combinatoire (NGA) augmente considérablement.

La connaissance des AI n'est pas suffisante pour pouvoir contrôler la performance du système robotique encore faut il pouvoir identifier les inducteurs de performance qui vont permettre de piloter la performance.

2.2.1.4 Principaux inducteurs de performance

Cette partie va présenter les principaux éléments impactant directement ou indirectement les performances d'un système robotique en mission autonome. Ces éléments peuvent découler des précédentes contraintes énoncées en rajoutant ceux considérées comme des éléments non prévisibles (perturbants ou pas).

a. Typologie

Il faut remarquer que la performance dans des missions robotiques dépend d'éléments relevant de domaines très différents. Une interdépendance est souvent observée. Cela va certainement rendre la tâche de management de la performance plus complexe. Les inducteurs de performance peuvent être proposées en fonction du contrôle que l'on peut exercer sur eux. Ils peuvent être :

- **Contrôlables hors ligne** : Certains inducteurs peuvent être paramétrés, ajustés ou intégrés avant le début de la mission. Une fois celle ci commencée, ces éléments ne sont plus modifiables : charge batterie, capacité de l'unité de calcul, efficacité algorithme, etc.
- **Contrôlables en ligne** : D'autres inducteurs sont modifiables en cours de mission. On peut agir sur ces derniers pour reconfigurer le système robotique en ligne de façon à ce que la mission reste réalisable tout en vérifiant les contraintes de performance imposées : vitesse, accélération, chemin, etc.

- **Non contrôlable** : Certains inducteurs ne sont pas pilotables que ce soit avant ou en cours de missions. Il sont généralement les caractéristiques des différents composants (fréquence d'acquisition et résolution de certains capteurs, consommation énergétique). Ils peuvent aussi provenir de sources de perturbations imprévisibles liées au dynamisme de l'environnement (turbulence, obstacle).

b. Classification

b.1. Environnement

C'est un paramètre crucial qui peut être au préalable connu ou pas et qui n'est pas nécessairement contrôlable. L'environnement peut être formé par des éléments statiques (murs, machines et tout élément immobile) et d'autres dynamiques composés d'objets inconnus temporairement statiques (chaise, table, voiture en parking), ou dynamique (homme, voiture en circulation).

Au sein d'un système robotique, l'environnement est décrit par un modèle qui transcrit/modélise que certains aspects de l'environnement réel. Ce modèle (graphe, surface, force, etc.) est construit via les données des capteurs dont le couple technologie/nature de l'environnement réel conditionne l'exactitude de ces données. La nature des éléments constituant l'environnement réel peut être alors d'une importance primordiale. Des propriétés telles que le coefficient d'absorption (couleur) et la matière (opacité) peut impacter considérablement l'exactitude des données des capteurs.

b.2. Ressources matérielles

-Capteurs

Ces sont les principaux éléments de perception de l'environnement. Ils impactent donc nécessairement fortement la performance en fonction de leurs caractéristiques technologie (fréquence, portée, principe) et leurs capacité à apercevoir les différentes nuances de l'environnement.

Leurs adaptation aux conditions environnementales dépend de la qualité et de l'exactitude du modèle construit de l'environnement. Le choix des capteurs et la qualité des données impacte donc directement la performance de localisation et la dimension sécuritaire de la mission. Par ailleurs, la nature de capteurs retenus pour exécuter une tâche robotique impacte aussi fortement la consommation énergétique du système robotique.

-Actionneurs

Ils sont contrôlables en ligne et ils permettent le déplacement dans l'environnement et l'interaction avec celui-ci. Ils sont bien évidemment indispensables pour assurer la mobilité des engins. Leurs caractéristiques techniques (couple, consommation, accélération, temps de réponse) vont là encore fortement impacter nombreux axes de performance comme l'énergie, la sécurité, la stabilité et la durée.

b.3. Ressources logicielles

-Algorithmes

Ils constituent le cœur des ressources logicielles d'un système robotique mobile. Ils peuvent être contrôlables en ligne s'ils sont paramétrables. Nous pouvons citer principalement des algorithmes de :

- Perception : À partir des données des capteurs, les algorithmes de perception fournissent des éléments d'observations de l'environnement (point plus proche, murs, porte) exploitables que se soit par le contrôle, localisation, détection, etc.
- Contrôle : Les algorithmes de contrôle commande le système robotique et définissent son comportement. Ils exploitent généralement les données issues des algorithmes de perception.

L'efficacité de ces algorithmes est généralement étudiée comme performance. Leurs caractéristiques comme les données d'entrées nécessaires, la charge de calcul et la mémoire, la fréquence, la qualité des données en sortie ou l'énergie consommée peuvent impacter les différents axes de performance.

-Architecture de contrôle

Elle gère les ressources matérielles et logicielles selon les contraintes et les objectifs de la mission. Elle sélectionne donc les algorithmes devant être utilisés tout au long de la mission en fonction des tâches robotiques à exécuter. Cela revient à choisir tout au long de la mission les ressources permettant de l'exécuter avec succès. D'autre part, l'architecture de contrôle est implémentée dans une unité de calcul qui consomme de l'énergie. Cette consommation dépend, entre autres, des algorithmes utilisés, de leur fréquence d'exécution et du processeur de l'unité de calcul. La fréquence de la boucle de contrôle est choisie aussi de façon à respecter les contraintes de stabilité et de réactivité du robot. La fréquence de contrôle et son paramétrage est donc un inducteur impactant directement la consommation énergétique du système robotique, sa capacité de réaction et donc la dimension sécuritaire, mais aussi bien évidemment la stabilité des lois de commandes implantées.

b.4. Source d'énergie

C'est une dimension cruciale pour le management de l'énergie d'un système robotique autonome. La réalisation d'une mission nécessite de disposer d'une réserve d'énergie suffisante pour mener à son terme. Les ressources énergétiques du système robotique peuvent être régénérées en utilisant des stations de recharge, panneaux photovoltaïques ou en récupérant l'énergie dissipée sous forme de chaleur, frottement, mouvement, etc. Les sources d'énergie pour un même système robotique peuvent être multiples et de différentes natures (thermique, électrique). Par exemple un ordinateur embarqué, une plateforme mobile et ses sous-systèmes (capteurs, actionneurs) peuvent disposer de sources d'alimentation distinctes.

Après avoir proposée une formalisation générique des principaux concepts liés à notre problématique nous allons maintenant présenter l'algorithme général de gestion des ressources.

2.2.1.5 Algorithme général de gestion des ressources guidée par la performance

Comme nous l'avons énoncé précédemment, une mission robotique se doit de respecter un ensemble de contraintes. Celles ci induisent des limitations aux niveaux des inducteurs de performance de mission.

Nous supposons à partir de maintenant que les inducteurs contrôlables hors ligne ont été identifiés, que leur nature, limitation sont connus et qu'ils permettent *a priori* de pouvoir envisager de réaliser la mission. En d'autres termes :

- La capacité énergétique du système robotique est connue.
- Ses capacités de perception et d'actionnement sont adaptées à la mission visée.
- Les algorithmes de contrôle/commande, perception, etc dont on dispose sont suffisants pour adresser les objectifs déclinés tout au long de la mission.

Nous nous intéressons maintenant aux inducteurs pilotables avant et en cours de mission. Ils correspondent à des configurations matérielles et logicielles (paramétrables ou pas) du système robotique. Elles ont été nommées alternatives d'implémentation paramétrées AI(P) (Définition 4).

Le but de l'approche proposée est de choisir localement, au niveau de chaque activité A_i composant la mission, une AI permettant globalement de mener la mission à son terme

tout en satisfaisant les contraintes de performance imposées par l'utilisateur. Ce choix doit être fait en respectant globalement (performance continue in fine comme l'énergie et la durée) et localement (performance booléennes comme la sécurité, localisation et stabilité) les différentes contraintes (mission et performance) envisagées. L'approche proposée permet de déterminer si a priori la mission est réalisable. Si c'est le cas, une solution *SAR* (Définition 9) permettant d'exécuter la mission sous les contraintes imposées est calculée.

Au cours de la mission, l'approche de gestion de ressources doit adapter dynamiquement et en temps réel la dernière solution trouvée si celle-ci n'est plus valable en raison des dérives expérimentales, des dysfonctionnements matériels ou logiciels ou de l'imprécision des modèles d'estimation utilisés.

Définition 9:

Une **Solution d'Allocation de Ressources SAR** est un plan prévisionnel des ressources utilisées (*AI*) pour réaliser chaque activité de la mission. Si la mission est réalisable, elle est initialement établie avant la mission (SAR_0). Une nouvelle solution est recalculée en cours de la mission (SAR_i) si la précédente (SAR_{i-1}) n'est plus applicable.

L'approche de gestion des ressources matérielles et logicielles repose sur 3 phases (Figure 2.8) [Jai+16c]. La première phase (P1) permet d'identifier les inducteurs de performance du système robotique étudié ainsi que les modèles d'estimation de performance correspondants. Le Scénario de Mission Détaillé SMD est établi en prenant en compte les contraintes de mission (légal, physiques/fonctionnelles et environnementales) et les contraintes de performance.

La deuxième phase (P2) est une étape d'estimation de la performance des différentes *AI* possibles basée sur le SMD établi en P1. Un algorithme de décision permet de chercher s'il existe une *SAR* permettant de mener à bien la mission tout en vérifiant l'ensemble des contraintes imposées.

Enfin, si la mission est réalisable, une phase d'évaluation en ligne de la performance (P3) vérifie tout au long de l'exécution de la mission si les performances réelles mesurées correspondent à celles prévues. Il est ainsi possible d'adapter dynamiquement et en temps réel la *SAR* pour poursuivre la mission en cours en satisfaisant toujours, si c'est possible les contraintes imposées. Les 3 phases de l'approche proposée vont maintenant être détaillées.

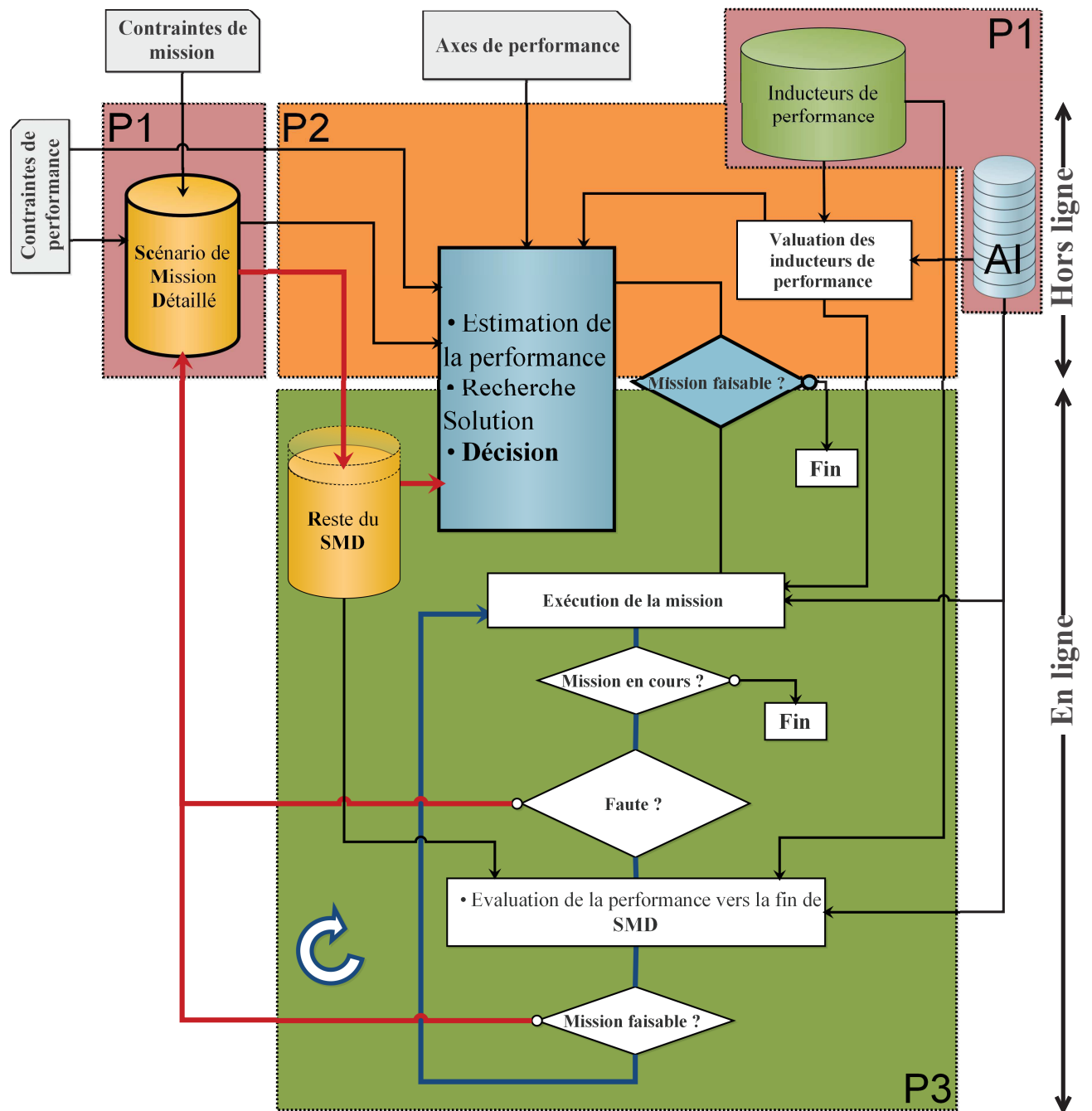


FIGURE 2.8 – Approche proposée de gestion des ressources guidée par la performance

2.2.2 Phase préliminaire (P1)

C'est la phase la plus importante. Elle se fait hors ligne en deux étapes. La première concerne l'identification des indicateurs et inducteurs de performance et la construction des modèles d'estimation correspondants. Une fois faite, il n'est plus nécessaire de la refaire si le système robotique reste inchangé. La seconde étape est l'identification du scénario de la mission.

2.2.2.1 Identification des indicateurs et inducteurs de performance et des modèles d'estimation associés

Cette étape de la phase 1 consiste à identifier les indicateurs et des inducteurs de performance pour chaque axe de performance. Les relations entre ces indicateurs et l'inducteur de performance sont modélisés par un diagramme causes-effets décrivant les liens causaux entre les éléments impactant le(s) indicateur(s) de performance (voir section 1.2.1.6). Il permet de représenter d'une manière synthétique et visuelle les relations entre les causes (inducteurs) aboutissant à un effet (indicateurs) sur la performance. Il est évident que l'utilisation de cet outil ne peut pas couvrir exhaustivement tous les inducteurs mais il permet de révéler les principaux éléments. D'autres outils peuvent être utilisés comme le diagramme d'interdépendance [BB03] qui, en comptabilisant le nombre des flèches entrantes et sortantes de chaque élément, on peut identifier les éléments les plus concernés et les plus influents.

La complexité de cette étape vient du fait que les inducteurs de performance sont généralement issus de différents domaines scientifiques et technologiques. Le défi est alors d'être capable d'estimer la performance de chaque AI et cela pour les différents axes de performance considérés.

Il faut donc établir les mécanismes ou les lois d'estimation de la performance pour chacun des indicateurs identifiés. Ces modèles peuvent être issus des normes, des caractéristiques cinématiques et dynamiques du robot, d'analyse probabiliste, empirique ou heuristique. Ces modèles d'estimation dépendent fortement du contexte d'étude et du robot étudié, de la finesse et de la justesse de l'analyse des graphes causaux et des modèles d'estimation de performance qui est indissociable de l'efficacité de l'approche proposée.

2.2.2.2 Construction du Scénario de Mission Détaillé (SMD)

L'estimation et la prédiction de la performance du robot nécessite forcément un scénario de mission connu. Ce scénario est établi en partant d'une description utilisateur de la

mission (Définitions 1 et 2). Cette description doit être enrichie par les informations issues des contraintes de mission et de performance d'une part, et d'autre part, par les fonctionnalités d'implémentation (ressources matérielles et logicielles) permettant d'exécuter les différentes activités constituant la mission (2.6).

Le passage de la description généralement multidimensionnelles à celle, unidimensionnelle, exploitable par les modèles d'estimation de la performance, se fait selon les étapes suivantes :

1. Définir le point de départ (début mission) comme l'origine du repère unidimensionnel (coordonnée 0).
2. Projeter linéairement sur la trajectoire planifiée les coordonnées des différentes zones de réalisation des **objectifs** et/ou des **contraintes** (Figure 2.6).
3. Définir l'espace spatio-temporel du déroulement des activités résultantes de l'intersection entre les objectifs et les contraintes. Ainsi on ajoute à chaque activité A_i les informations suivantes : une coordonnée linéaire de départ x_{A_i} , une distance parcourue par le système robotique durant l'activité considéré dis_{A_i} **ou** une durée fixée ou estimée dur_{A_i} récapitulés dans le vecteur d'informations I_{A_i} . Cette décomposition spatio-temporelle s'imposera à chaque AI d'une activité.

$$I_{A_i} = \begin{pmatrix} x_{A_i} \\ dis_{A_i} \\ dur_{A_i} \end{pmatrix} \quad (2.9)$$

4. Appliquer les contraintes de mission et d'environnement sur les fonctionnalités d'implémentation consiste à :
 - Filtrer les AI qui restent éligibles/applicables par rapport aux contraintes (ie. capteurs ou algorithmes non fiables dans une zone).
 - Borner ou déterminer les valeurs possibles de l'ensemble P de ses paramètres pour chaque AI d'une activité (e.g. limiter la vitesse d'un robot ou exiger une fréquence minimale d'un capteur).

Étapes de construction d'un SMD

Un **Scénario de Mission Détaillé SMD** est ainsi établi (2.10). Il contient les détails sur les choix de ressources (ensemble des AI et les coordonnées / distance / durée associées) pour la réalisation de la mission tout en respectant les contraintes de missions et de perfor-

mance. On associe alors à chaque activité A le couple (ensemble des AI , informations sur la déroulement spatio-temporel de l'activité récapitulé dans I (2.9)). L'ensemble d' AI par activité ainsi que l'ensemble de paramètres P doit vérifier les contraintes de performance booléennes (sécurité et localisation).

$$SMD = \{A_1^{(\{AI(P)\}_{A_1}, I_{A_1})}, A_2^{(\{AI(P)\}_{A_2}, I_{A_2})}, \dots, A_x^{(\{AI(P)\}_{A_x}, I_{A_x})}, \dots, A_{n_{act}}^{(\{AI(P)\}_{A_{n_{act}}}, I_{A_{n_{act}}})}\}_{seq} \quad (2.10)$$

Un exemple générique du processus de création du SMD est présenté dans la figure 2.9. La mission considérée est une séquence de deux objectifs de déplacement O_I et O_{III} et un objectif sans déplacement à durée fixe $dur_{O_{II}}$. Les coordonnées de départ de O_I et O_{III} sont indiquées et les durées ne sont pas connues puisque la vitesse n'est pas encore choisie. Deux contraintes sont appliquées sur cette mission. Elles sont interprétées par des restrictions sur l'utilisation de certains AI (AI non valable *n.v.*) et des limites maximales ou minimales imposées pour certains paramètres. Les contraintes peuvent être considérées ou pas selon les axes de performance pris en compte. Supposant que la condition AI_3 est *n.v.* dans c_2 pour des raisons sécuritaires, elle ne sera prise en compte que si l'axe sécuritaire est considéré (ce qui indispensable dans la réalité mais peut être négligé en simulation par exemple). Par ailleurs, la seule application des contraintes sur les objectifs de la mission peut la rendre irréalisable si ces dernières se chevauchent tout en étant antinomiques.

La figure 2.9 montre le découpage en activités résultant de l'application des contraintes c_1 et c_2 ainsi que les ensembles de AI **éligibles** et leurs paramètres correspondants. On peut remarquer que A_I et A_{VII} garde le même ensemble d' AI que leurs objectifs correspondants O_I et O_{III} . En revanche, pour les autres activités la cardinalité des $\{AI\}$ diminue en raison de la projection des contraintes imposées.

Les informations sur la déroulement spatio-temporel de chacune des activités sont récapitulées dans le tableau 2.2. Nous retrouvons les vecteurs I_{A_i} d'information de chaque activité A_i . La coordonnée linéaire de départ de la première activité est bien évidemment égale à 0. De plus nous pouvons remarquer que l'activité statique A_{IV} est statique puisque la distance à parcourir correspondante est nulle tandis qu'elle a une durée prédéfinie.

Tableau 2.2 – Les valeurs du vecteur I pour l'exemple de la figure 2.9

	I_{A_I}	$I_{A_{II}}$	$I_{A_{III}}$	$I_{A_{IV}}$	I_{A_V}	$I_{A_{VI}}$	$I_{A_{VII}}$
x_{A_i}	$x_{O_I} = 0$	x_{c_1}	x_{c_2}	$x_{O_{II}} = x_{O_{III}}$	$x_{c_1} + dis_{c_1}$	$x_{c_2} + dis_{c_2}$	
dis_{A_i}	$x_{A_{II}} - x_{A_I}$	$x_{A_{III}} - x_{A_{II}}$	$x_{A_{IV}} - x_{A_{III}}$	0	$x_{A_{VI}} - x_{A_V}$	$x_{A_{VII}} - x_{A_{VI}}$	$dis_{O_I} + dis_{O_{III}} - x_{A_{VII}}$
dur_{A_i}	-	-	-	$dur_{O_{II}}$	-	-	-

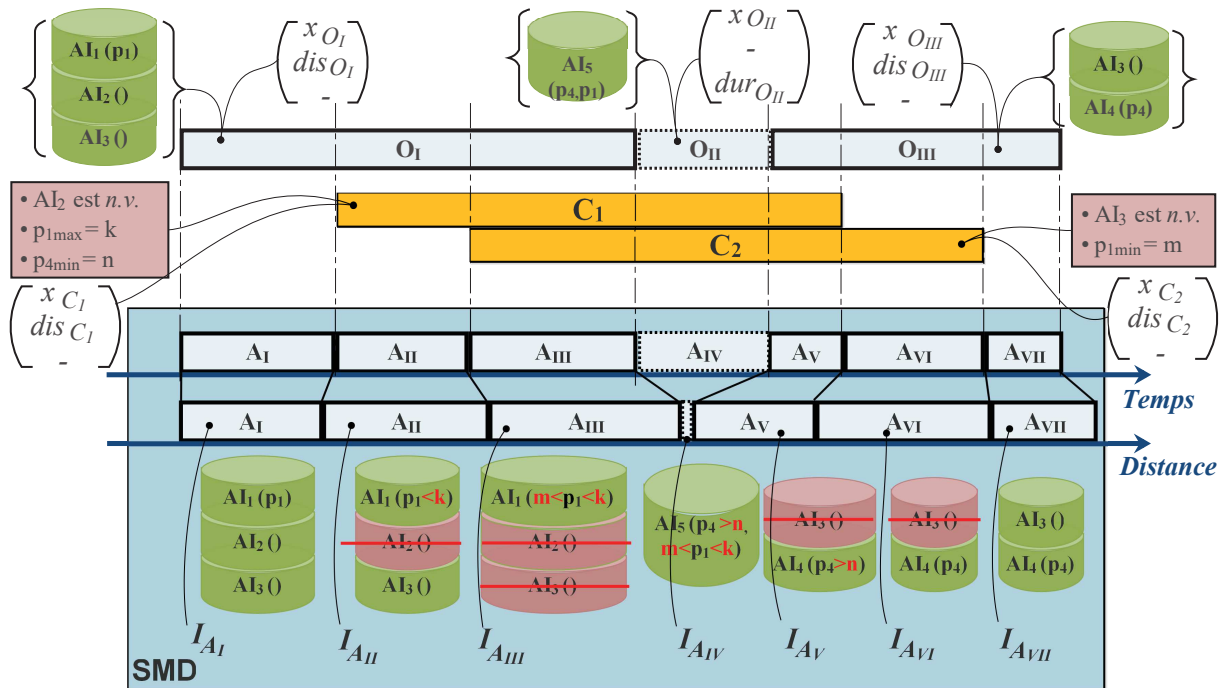


FIGURE 2.9 – Exemple de création d'un SMD

Nous avons maintenant identifié, pour chaque axe de performance les indicateurs et les inducteurs de performance ainsi que les modèles et mécanismes d'estimation de performance associés.

Un scénario détaillé SMD de la mission prenant en compte les contraintes imposées a été construit.

Une fois établi, il faut remarquer que si le SMD est composé d'un ensemble non nul, toutes les contraintes de performance booléennes sont vérifiées.

Il faut maintenant estimer la performance au niveau des activités à la mission. Celle ci va dépendre des ressources matérielles et logicielles choisies et des paramètres qui leur seront associés. Un processus de décision doit donc maintenant être engagé pour chercher une solution *SAR* permettant de réaliser la mission en prenant en compte les contraintes globales de performance (performance terminale comme l'énergie et la durée) imposées. Il est donc maintenant nécessaire d'estimer la performance de mission hors ligne pour conclure ou non sur la faisabilité de la mission. C'est le rôle de la seconde phase de l'approche de gestion des ressources (Phase P2).

2.2.3 Estimation hors ligne de la performance (P2)

Dans un premier temps cette seconde phase consiste à estimer si la mission est réalisable ou non. Elle doit en fonction du SMD établi, des modèles d'estimation de performance de chaque indicateur de performance et des contraintes de performance terminale fixée, estimer les performances qui devraient être observées au niveau des activités de la mission en fonction des ressources choisies et de leur paramétrage. Cette première étape est purement locale pour chaque AI d'une activité. C'est dans un second temps que l'aspect global de la performance est pris en compte pour choisir les ressources à engager pour chaque activité de façon à satisfaire les contraintes globales de performance imposées. S'il existe une solution d'allocation de ressources elle correspond à *SAR* (Définition 9).

Le processus de décision est la clef de voûte de l'approche proposée. Il sera par ailleurs utilisé si nécessaire en ligne dans la phase P3 de notre démarche. La problématique de décision peut être formalisée en un problème sac à dos connu pour être NP-complet. En effet la complexité combinatoire de *NGA* (2.8) augmente rapidement et peut devenir colossale. Cela exclue de considérer directement toutes les combinaisons d'allocations puisque même les algorithmes de classement les plus efficaces ne pourraient classer les performances associées à l'ensemble d'état. Ainsi, explorer exhaustivement l'espace des combinaisons possibles d'alternatives d'implémentation devient impossible en temps réel. Beaucoup d'algorithmes ont été proposés pour répondre à ce genre de problème (voir 1.3.2.2). Nous nous sommes appuyés sur celui proposé au LIRMM par [Ben+05] pour l'allocation des ressources humaines aux processus d'entreprise. Cet algorithme doit être tout de même adapté à notre contexte et domaine d'utilisation.

2.2.3.1 Formalisation sac à dos du problème d'allocation de ressources

Rappelons que la contrainte de performance de l'indicateur ind_i selon un *axe* considéré est notée $Per f_{C,axe}^{ind_i}$. On s'intéresse bien évidemment aux contraintes continues qui doivent être respectées *in fine* puisque les contraintes booléennes sont nécessairement respectées une fois le *SMD*₀ établi. On considère alors :

- un ensemble d'activités $A = \{A_j / 1 \leq j \leq n_{act}\}$.
- pour chaque activité A_j , l'ensemble possible de Choix Locaux AI, $CL_{A_j} = \{AI^{j_k}\}$ avec $1 \leq j_k \leq n_{AI}^{A_j}$ (Figure 2.7).
- *Obj* est la contrainte de performance globale $Per f_{C,axe}^{ind_i}$.
- pour chaque activité, une fonction d'évaluation (modèle d'estimation) f_i , permet de calculer la valeur correspondante de performance pour chaque $AI_{A_j}^{j_k}$.

- une variable booléenne $x_{A_j}^{j_i} \in \{0, 1\}$ associée au choix local $AI_{A_j}^{j_k}$. Au niveau de chaque activité, un choix unique peut être fait de façon à ce que :

$$\sum_{j_k=1}^{j_k=n_{AI}^{A_j}} x_{A_j}^{j_k} = 1 \quad (2.11)$$

- La combinaison de choix locaux maximisant au mieux l'objectif global de performance devant être trouvée, doit vérifier (2.12).

$$\sum_{i=1}^{i=n_{act}} \sum_{j_i=1}^{j_i=n_{AI}^{A_j}} x_{A_j}^{j_k} \cdot f_i \left(AI_{A_j}^{j_k} \right) \leq Obj \quad (2.12)$$

Cette formulation est celle d'un problème de sac à dos. La formulation générale de ce problème étant définie, la suite va détailler l'algorithme permettant de trouver une bonne solution tout en respectant les contraintes temps réel d'implémentation informatique.

2.2.3.2 Algorithme

L'algorithme défini dans [Ben+05] présente une solution efficace au problème de sac à dos (Fig. 2.11). Dans ses travaux, Bennour a prouvé que l'algorithme proposé permet de trouver une bonne (pas forcément la meilleure) solution, si elle existe, plus rapidement que les approches de Pisinger [Pis95], *Hybrid Taboo-Descent* [HGH99] ou *recuit-simulé* [Met+53]. Il suppose tout d'abord qu'on dispose d'un ensemble fini et discret de valeurs de performance au niveau de chaque activité.

Son efficacité repose d'une part sur des listes classées (ordonnées) des choix locaux $AI_{A_j}^{j_k}$ au niveau des activités (Figure 2.10). D'autre part, il suppose que la loi de composition de performance globale préserve le classement local (e.g. addition pour les axes de durée et énergie). Alors en partant d'une sélection initiale locale pour chaque activité, l'algorithme utilise une simple recherche dichotomique (complexité $O(\ln(n))$) pour trouver une solution satisfaisant l'objectif de performance (énergie et durée) fixé. Le problème NP-complet est donc résolu par un algorithme de complexité linéaire (temps/espace).

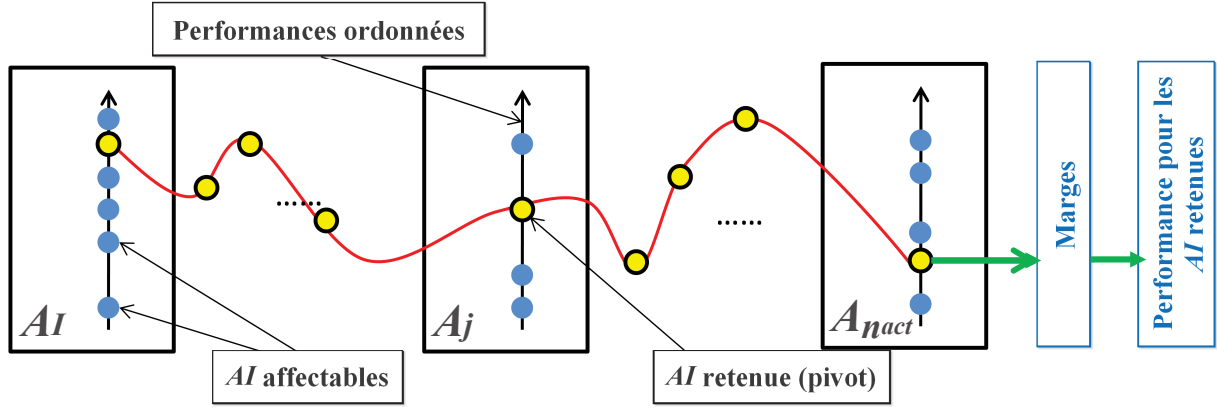


FIGURE 2.10 – Formalisation sac à dos multicritère

Il est évident que le nombre de valeurs possibles, pour les axes de performances dont les lois sont continues (e.g. durée et énergie), est infini. Il faut alors discrétiser des lois afin d'avoir un nombre fini de valeurs et pouvoir effectuer un choix [PBR93]. Pour y parvenir, deux stratégies sont adoptées :

- **Stratégie 1 (Axe durée)** : minimiser la durée de la mission. Elle correspond à adopter au niveau de chaque activité de la mission la vitesse maximale possible.
- **Stratégie 2 (Axe énergie)** : maximiser la consommation énergétique **tout en respectant la contrainte imposée**. Ceci revient généralement à maximiser le nombre de ressources utilisées et correspondent aux configurations matérielles et logicielles les plus performantes.

L'algorithme proposé est illustré dans la figure 2.11. Localement, pour une activité A_j un choix de pivot AI_{PI} (point jaune la figure 2.10) $AI_{A_j}^{P_k}$ divise l'ensemble des AI en deux sous-ensembles ayant des performances supérieures et inférieures à celle de $AI_{A_j}^{P_k}$ grâce à la classification locale des AI . La prise en compte des AI sélectionnées pour toutes les activités de la mission permet de calculer globalement la performance de cette combinaison (Figure 2.10). Les choix de pivots AI_{PI} est modifié par la suite activité par activité en utilisant un algorithme dichotomique jusqu'à trouver une bonne solution respectant *Obj*. La complexité NP-complet du problème sac à dos est réduite ainsi à la complexité $(k \cdot O(\ln(n_i)))$ d'un problème dichotomique, ce qui le rend résoluble en temps réel en raison du faible nombre d'itérations nécessaires pour trouver une solution si elle existe. Avec k est le nombre d'activités et n_i est le nombre d'AI pour l'activité i .

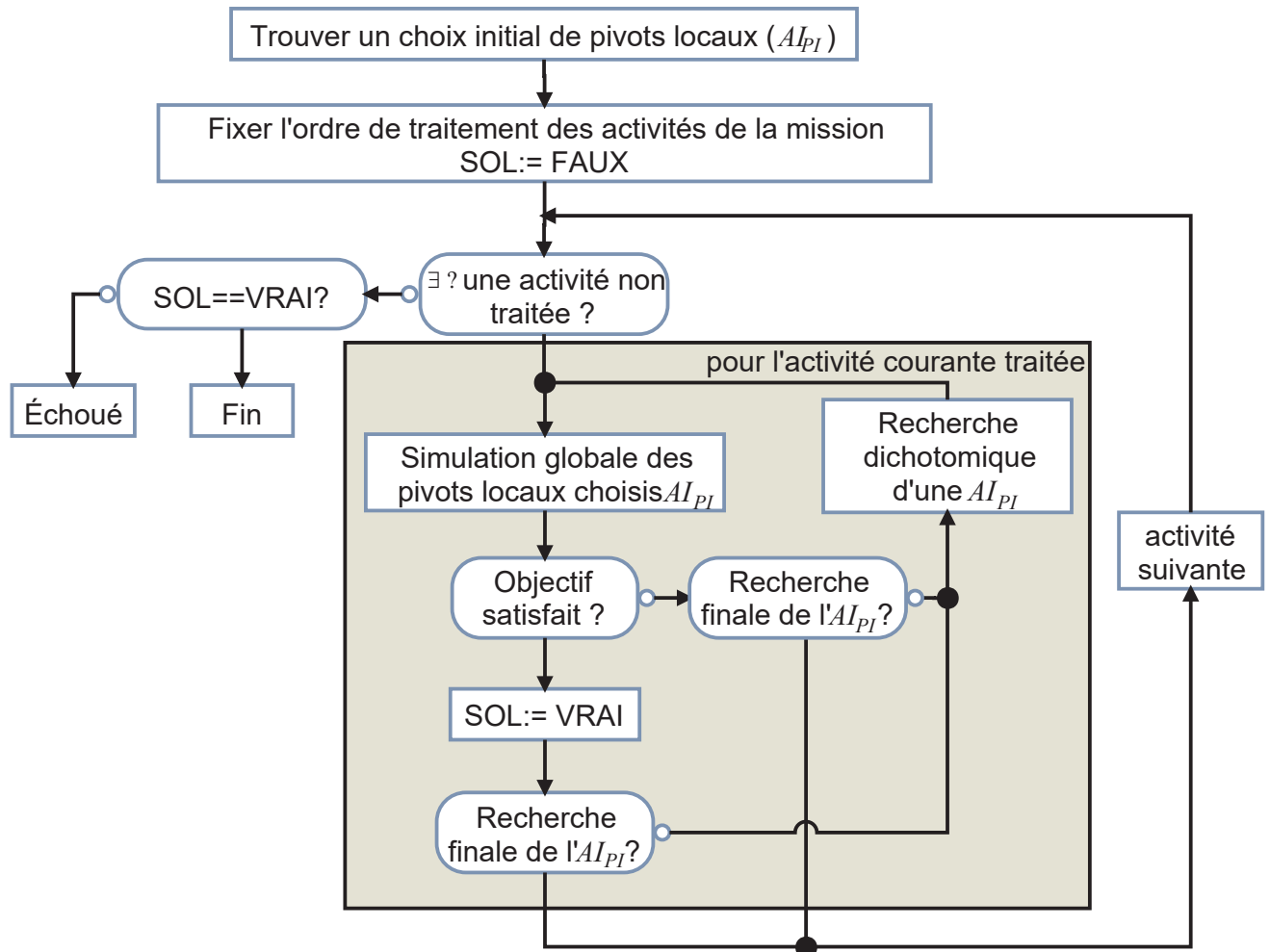


FIGURE 2.11 – Algorithme de recherche dichotomique proposé par [Ben+05]

Rappelons que le problème à résoudre est multicritère puisque plusieurs axes de performance doivent être considérés simultanément. Dans ces travaux, Bennour propose d'adresser cette problématique en utilisant de façon itérative l'algorithme présenté précédemment. À l'issue du traitement de chaque point de vue (indicateur) le sous ensemble de solutions résultant sera l'ensemble d'entrée du même algorithme mais du point de vue suivant. L'ensemble de solutions finales satisfait donc nécessairement tous les points de vue traités précédemment. Il faut remarquer que l'ordre de traitement des points de vue impact la solution finale. Par exemple, supposant deux indicateurs de performance ($A = \max(x)$ et $B = \min(y)$). Si on traite A puis B, la solution sélectionnée va respecter la contrainte A tout en minimisant B. Le cas contraire, la solution sélectionnée respectera B en maximisant A. Cela montre que l'ordre de traitement de point de vue peut donc avoir une importance non négligeable. Cet ordre sera choisi heuristiquement par l'utilisateur en fonction de son retour d'expérience.

À l'issue de la phase 2, si une SAR_0 est identifiée par l'algorithme d'allocation alors la mission peut commencer. La phase 3 de notre méthodologie qui permet d'estimer en ligne la réalisabilité de la mission au cours de sa progression peut alors être engagée. En revanche, s'il n'existe pas une solution alors la mission n'est pas réalisable sous les contraintes de performance imposées.

2.2.4 Estimation et gestion en ligne de la performance (P3)

SAR_0 a été déterminée à partir de modèles de performance estimés en supposant l'environnement statique et que toutes les fonctionnalités du robot sont opérationnelles. La mission peut commencer en appliquant cette solution. Cependant, la réalité du terrain peut être très différente de celle prévue. Cette différence amène des dérives de performance. Elles peuvent être dues à :

- des erreurs des modèles d'estimation des performances en raison des différences entre les hypothèses et la réalité (vitesse et/ou accélération constante, trajectoire suivie différente de celle prévue, etc.).
- des perturbations du chemin et de la trajectoires de système robotique dues aux obstacles non prévus (évitement d'obstacles).
- des fautes inopinées sur des ressources matérielles ou logicielles.

Typologie des fautes considérées

De ce fait, à partir du plan estimé en P2 et durant l'exécution de la mission, les performances réelles vont être périodiquement évaluées et comparées aux performances estimées pour décider si la mission reste réalisable ou non. Les marges de performance continues et booléennes sont donc périodiquement contrôlées. Les marges de performance continues constituent autant de réserves de performance qui nous permettront d'absorber les imperfections de nos modèles et le dynamisme imprévu de l'environnement.

De plus, une faute sur une ressource matérielle ou logicielle peut disqualifier une AI courante ou planifié pour être utilisée ultérieurement. Le dynamisme de l'environnement peut causer des évitements d'obstacles imprévus ce qui va rallonger la trajectoire du robot et ainsi provoquer un comportement non prévu amenant, entre autres, à des pertes de temps et d'énergie. Cette perte peut mener à des marges de performance *in fine* négatives. Cela signifie que les contraintes de performance de la mission ne peuvent plus être respectées avec la SAR_i courante.

Une nouvelles SAR_i doit être calculée en raison de l'un ou plusieurs des facteurs suivants :

- Environnement sans imprévus mais il y a une erreur de prédiction pour les modèles d'estimation utilisés qui conduit à une(des) marge(s) de performance négative(s).
- Dynamisme dans l'environnement qui engendre une dérive de performance et conduit à une(des) marge(s) de performance négative(s).
- Une AI courante ou planifiée n'est plus utilisable en raison d'un dysfonctionnement matériel ou logiciel.

Pour palier à cette situation, le processus de recherche de décision utilisé en P2 est réutilisé en ligne sur le **Reste du Scénario de Mission Détaillé $RSMD$** (Figure 2.8). Il correspond à la description détaillé du reste de la mission à partir de la position spatio-temporelle courante du robot dans sa mission.

Grâce à l'efficacité de l'algorithme d'allocation choisi, il cherche en temps réel une nouvelle SAR_{i+1} permettant de réaliser les activités restantes prenant en compte les performances courantes mesurées. La mission peut être poursuivie si une SAR_i satisfaisant les contraintes de performance (booléennes et continues) est trouvée. Dans le cas contraire, la mission est abandonnée (ou déviée vers une autre objectif comme retour à la base, recherche zone sécurisée, etc.) et l'utilisateur est prévenu.

2.3 Conclusion approche de gestion des ressources

En premier lieu, nous avons proposé dans ce chapitre, une formalisation du concept de performance dans le cadre de la robotique mobile. Ce concept est souvent utilisé en robotique mobile mais contrairement au domaine de la productique, il n'a pas été clairement formalisé. Nous avons alors défini deux classes de performance qui permettent de distinguer les performances qui sont indispensables pour l'exécution d'une mission robotique mobile (axes principaux) de ceux qui servent à évaluer la qualité du service rendu (axes secondaires). Nous avons détaillé dans cette partie les notions liées à la performance comme la mesure, l'estimation et l'évaluation. La notion de marge de performance a été également proposée. Elle permet de juger de la réalisabilité de la mission avant et en cours de la mission. Elle caractérise lorsqu'elle est positive une réserve de performance qui permet d'absorber les impacts dûs aux approximations des modèles et au dynamisme de l'environnement.

La deuxième partie de la méthodologie proposée consiste à détailler la démarche de gestion des ressources guidée par des contraintes de performance. Cette méthodologie

permet de faire un choix objectif des ressources devant être utilisées tout au long d'une mission robotique en vérifiant des performance exigées. Les principaux types et classes d'inducteurs performance impactant les performances d'un système robotique ont été aussi présentés. Parmi ceux certains sont contrôlables (ou non), avant et/ou en cours de la mission.

Nous avons ensuite détaillé l'algorithme de gestion des ressources qui permet de décider de la réalisabilité d'une mission robotique vis à vis des différentes contraintes de performance. Cet algorithme se compose de 3 principales phases. La première consiste à identifier les inducteurs de performance ainsi que les indicateurs de performance et leurs modèles d'estimation correspondants. Elle permet aussi de construire une description détaillée de la mission (SMD) prenant en compte des différentes contraintes de performance booléennes la mission. Cette description représente une décomposition de la mission en activités supportant des contraintes invariantes. Une description spatio-temporelle est alors associé à chacune activité ainsi que l'ensemble de AI éligibles permettant de l'exécuter. La deuxième phase estime si la mission est réalisable en calculant une SAR_0 respectant les contraintes de performance continues. Cette SAR correspond à la solution d'un problème sac à dos multicritère résolu en utilisant l'algorithme dichotomique [Ben+05]. Ce dernier permet de trouver une bonne solution en temps réel malgré la complexité du problème. Enfin, au cours de l'exécution de la mission, la dernière phase consiste à observer périodiquement les performances réelles et à mettre à jour les marges de performance afin de vérifier si la mission reste réalisable et à recalculer une nouvelle SAR si nécessaire.

La démarche proposée est applicable sur les missions robotiques mobiles autonomes dont le scénario est connu. Pour valider cette méthodologie en simulation et expérimentalement, nous allons la projeter dans la suite de ce manuscrit sur une mission autonome mobile terrestre. Cela permettra d'explicitier et d'opérationnaliser l'approche proposée et de clarifier certains points encore trop abstraits. Pour cela, le contexte expérimental de la mission robotique choisie et le système robotique mobile terrestre utilisé vont être présentés dans le prochain chapitre.

Chapitre 3

Contexte applicatif et expérimental

Nous avons présenté dans le deuxième chapitre l’approche de gestion des ressources matérielles et logicielles guidée par la garantie de performance. Cette approche est applicable pour différents systèmes autonomes mobiles et pour une mission robotique où un scénario nominal peut être défini spatialement et/ou temporellement. Cependant sa mise en œuvre dépend considérablement du système étudié, de sa technologie, de sa conception, de l’environnement dans lequel il évolue, etc. Ce chapitre va présenter l’architecture matérielle et logicielle du système robotique ainsi que la mission cible sur lesquelles l’approche de gestion de ressources va être déployée.

Sommaire

3.1	Plateforme expérimentale	83
3.1.1	Ressources matérielles	83
3.1.2	Ressources logicielles	87
3.2	Mission de patrouille (<i>patrolling</i>)	101
3.2.1	Description de la mission	101
3.2.2	Environnement	102
3.2.3	Contraintes de performance	103
3.3	Conclusion	104

3.1 Plateforme expérimentale

3.1.1 Ressources matérielles

Le système robotique mobile utilisé pour les expérimentations est composé d’une base mobile Pioneer 3DX sur laquelle une structure en aluminium a été ajoutée pour accueillir

l'ordinateur de contrôle et les capteurs embarqués ajoutés (Figure 3.1).



FIGURE 3.1 – Système robotique mobile d'expérimentation

Les différents composants sont décrits dans la suite.

3.1.1.1 Plateforme mobile de base

Le Pioneer 3DX¹ est un robot non holonome de type unicycle produit par *adept MOBILEROBOTS* qui est largement utilisé en recherche. Le robot pèse 9 kg avec une charge utile de 17 kg (Figure 3.2). Théoriquement il peut atteindre la vitesse $v_{r_{max}}^{th} 1.2 m/s$ mais expérimentalement le système utilisé est limité à $v_{r_{max}}^{exp} 0.76 m/s$.

Il dispose d'un contrôleur Hitachi H8S qui gère ses deux moteurs et les sonars. Ces derniers sont disposés en deux banques (avant et arrière). Chacune contient 8 capteurs avec un angle de 20° entre les directions de deux capteurs consécutifs. Leurs gains ainsi que leurs fréquences d'insonification (onde) peuvent être ajustés. Le Pioneer dispose accessoirement de 5 *Bumpers* à l'avant et à l'arrière permettant de détecter un contact physique avec l'environnement. Le robot est aussi équipé de deux encodeurs sur ses deux roues qui permettent d'estimer la pose odométrique du robot ainsi que sa vitesse.



FIGURE 3.2 – Base mobile Pioneer 3DX

1. <https://www.cyberbotics.com/guide/using-the-pioneer-3-at-and-pioneer-3-dx-robots.php>

Le robot est contrôlé par un ordinateur embarqué via une communication série USB. Après établissement de la communication, l'ordinateur de commande peut envoyer des requêtes (vitesses roues, paramètres PID pour la régulation des vitesses, fréquence sonars, etc) et recevoir des informations sur le robot (encodeurs, tension batterie, données sonars, etc). Tous les équipements du robot sont alimentés par trois batteries au plomb 12 V et totalisant une charge énergétique théorique de 252 Wh.

3.1.1.2 Éléments ajoutés

Bien que la plateforme du Pioneer fournisse les fonctions de base d'un robot mobile, l'exploiter pour des applications relativement avancées dans des environnements complexes nécessite l'ajout de capteurs extéroceptifs et périphériques additionnels et d'un ordinateur embarqué qui héberge les différents drivers et algorithmes.

a. Capteurs

- En plus des sonars fournis avec le Pioneer 3DX, deux scanners laser Hokuyo² ont été ajoutés (Figure 3.3). Ils couvrent chacun 240° avec une portée de 4 mètres. Les deux lasers montés tête bêche, permettent de couvrir les 360° autour du robot. La période d'un balayage est de 100 ms avec une résolution de 0.36°. Les caractéristiques du laser sont nettement meilleures que celles des sonars. Cependant il faut remarquer la faible sensibilité des faisceaux laser aux objets opaques et transparents qui sont mal ou non détectés. Ils sont donc complémentaires avec les sonars.



FIGURE 3.3 – Scanner laser Hokuyo URG-04LX

- Le capteur Kinect^{©3} est très utilisé en robotique (Figure 3.4). Il permet à la fois de réaliser des captures d'image 3D, d'effectuer de la reconnaissance faciale, gestuelle et vocale et de capturer des flux vidéos. Il se compose d'un capteur de profondeur

2. https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html

3. <http://www.xbox.com/en-US/xbox-360/accessories/kinect>

composé d'une matrice de points infrarouges projetés sur la scène et d'une camera RGB, d'un microphone et d'une base motorisée. Dans le cadre de cette thèse seule la fonctionnalité de capture de vidéo est employée.



FIGURE 3.4 – Camera Kinect[©]

b. Périphériques

D'autres périphériques ont été ajoutés pour pouvoir gérer l'activation des ressources matérielles et suivre la consommation énergétique du robot.

- Deux cartes contenant chacune deux relais permettant de gérer l'alimentation des lasers et de la Kinect. L'alimentation des relais peut se faire en externe (5 V) ou par la connexions USB (Figure 3.5).

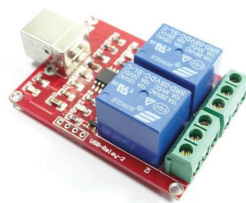


FIGURE 3.5 – Relais USB-RELAY 2

- Le module wattmètre Yocto-Watt⁴ mesure la consommation électrique en sortie de la batterie du robot (Figure 3.6). Il fournit différentes mesures électriques telles que le courant, la tension, la puissance et l'énergie consommée (précision 1 mWh) ainsi que la durée de mesure. Ces mesures sont communiquées vers le PC embarqué via une connexion USB. Elles peuvent être également enregistrées sur une mémoire flash.

4. <http://www.yoctopuce.com/EN/products/usb-electrical-sensors/yocto-watt>

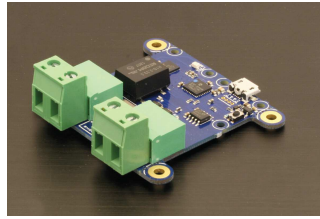


FIGURE 3.6 – Wattmètre Yocto-Watt

c. Ordinateur embarqué

La communication avec les différentes ressources matérielles présentées précédemment est faite par un ordinateur embarqué. Il dispose d'un processeur Intel i5 supportant un système d'exploitation Linux-RTAI. Il a sa propre batterie assurant son alimentation. Elle peut fournir jusqu'à 87 Wh. Les drivers, les algorithmes de contrôle et de perception ainsi que l'approche de gestion des ressources sont implémentés sur cet ordinateur en langage C/C++, grâce à une architecture de contrôle que nous allons maintenant présenter.

3.1.2 Ressources logicielles

Le contrôle du robot, l'acquisition des données capteurs ainsi que la mise en œuvre de l'approche de gestion des ressources du robot dans le but de garantir des performances nécessite l'utilisation d'une architecture modulaire temps-réel. Nous nous sommes appuyés dans le cadre de cette thèse, sur le *Middleware ContrACT* (Control Architecture Creation Technology) développé au LIRMM, qui permet de simplifier la conception de l'architecture logicielle. Ces grands principes sont exposés dans la section suivante.

Nous allons dans un premier temps présenter les principes et concepts qu'il déploie avant de décrire dans un second temps les modules fonctionnels utilisés dans le cadre de notre étude.

3.1.2.1 Architecture de contrôle

L'architecture de contrôle utilisée est générée par le *Middleware ContrACT*. C'est un outil développé au LIRMM facilitant le déploiement d'architecture de contrôle temps-réel [Pas10]. L'architecture conçue par ContrACT présente une décomposition générique à base de modules décisionnels (superviseur) pour le contrôle, et de modules exécutifs pour les fonctions. L'architecture est indépendante du système cible et est constituée par des briques logicielles appelées modules. Une association de modules constitue un schéma.

a. Module

Le module ContrACT est la brique de base et peut être divisée en deux parties illustrées dans la figure 3.7.

- Un code généré par ContrACT permettant de gérer le comportement du module : création, destruction, lien avec l'extérieur (réception des données et d'événements), appel au code interne, etc.
- Un code "utilisateur" définissant la fonctionnalité du module et décrivant son comportement interne.

Un module ContrACT peut être soit **synchrone** avec une exécution périodique (drivers des capteurs, contrôle du robot, etc) ou **asynchrone** (apériodique) où son exécution est amorcée par un événement externe (superviseurs, gestion des alimentations des capteurs, etc).

L'interface qui permet d'échanger des informations entre les modules ou de paramétrer leur comportement peut comporter différents types d'interactions :

- **Port de requête** : Point d'entrée de messages permettant de contrôler l'activité d'un module (démarrage, arrêt) et sa configuration (fixation d'un paramètre, abonnement de ses flux).
- **Port de paramètres** : Point d'entrée permettant de fixer la valeur d'un paramètre "public".

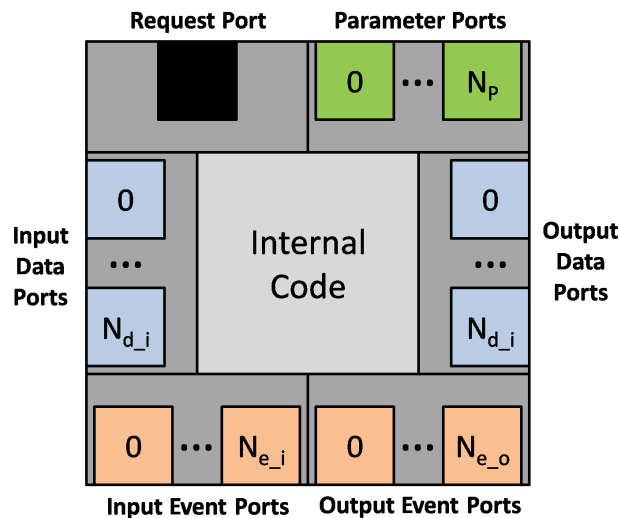


FIGURE 3.7 – Schéma d'un module (figure tirée de [Pas10])

- **Port d'entrée de données** : Il correspond à un flux de données consommées par le module. Il peut être connecté à des ports de sortie de données d'autres modules (module périodique).

- **Port de sortie de données** : Il correspond à des flux de données produites par les modules. Ce port peut être connecté à des ports d'entrée de données d'autres modules (module périodique).
- **Port d'entrée d'événements** : Il correspond à des flux d'événements entrant dans le module. Ce port peut être connecté à des ports de sortie d'événements d'autres modules (module périodique ou aperiodique).
- **Port de sortie d'événements** : Il correspond à des flux d'événements générés par les modules. Ce port peut être connecté à des ports d'entrée d'événements d'autres modules (module périodique ou aperiodique).

b. Schéma

Les modules périodiques sont exécutés dans des schémas à périodes fixes (Figure 3.8). On peut trouver au sein d'un même schéma un ou plusieurs modules reliés (ou pas) par des flux de données. Leurs exécutions peuvent être contraintes par des règles de précedence.

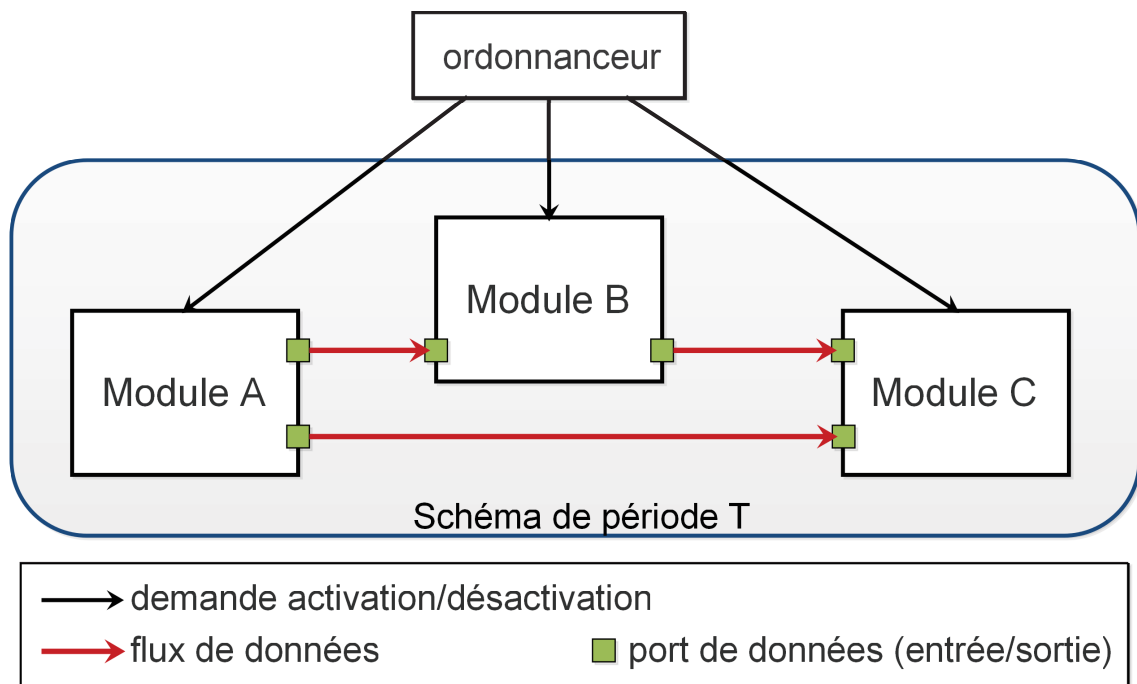


FIGURE 3.8 – Structure d'un schéma ContrACT

Bien évidemment, plusieurs schémas peuvent fonctionner en parallèle et à différentes fréquences.

c. Structuration de l'architecture

L'architecture contenant ces différents modules et schémas est structurée en deux couches (niveaux) (Figure 3.9)

- **Couche décisionnelle** : Elle gère la prise de décision et les réactions aux événements provenant de la couche exécutive. Les modules appartenant à cette couche sont appelés superviseurs et sont organisés hiérarchiquement. Ils fonctionnent d'une façon événementielle aperiodique. Les superviseurs interagissent entre eux et avec les modules des couches décisionnelles hiérarchiquement inférieures ou de la couche exécutive. Ils assurent les traitements liés à la prise de décision en fonction du contexte. Ils définissent les réactions d'activation/désactivation des schémas et les paramétrages des modules.
- **Couche exécutive** : Elle contient les modules périodiques ou aperiodiques réalisant les décisions prises dans la couche décisionnelle. Ces modules contiennent les traitements de calcul et de mémorisation, les algorithmes de commande et de perception, le codage des mécanismes d'interactions avec l'extérieur (e.g. communication réseau, capteurs, actionneur). On distingue les modules périodiques et événementiels :
 - Le domaine synchrone représente les modules exécutés périodiquement et le module ordonnanceur qui contrôle leur exécution.
 - Le domaine asynchrone contient les modules qui s'exécutent suite aux demandes des superviseurs. Ils peuvent être exécutés en priorité ou en temps restant.

Les différents modules fonctionnels de la couche exécutive utilisés dans le cadre de ce manuscrit vont maintenant être décrits. On y trouve des modules d'interfaçage entre l'architecture logicielle et l'architecture matérielle (drivers) ainsi que tous les modules déployant les algorithmes de contrôle, localisation, perception, traitement des données issues des capteurs, etc.

3.1.2.2 Modules fonctionnels

a. Module P3D : Gestion de la plateforme du robot Pioneer-3DX

C'est un module périodique qui permet de faire la communication et le contrôle de la plateforme mobile. Il assure la liaison de l'architecture avec le robot via une liaison série USB. Cette communication supporte l'envoi de commandes (e.g. vitesses des roues gauche et droite, fréquence des sonars, paramètres PID du régulateur de vitesse) et la

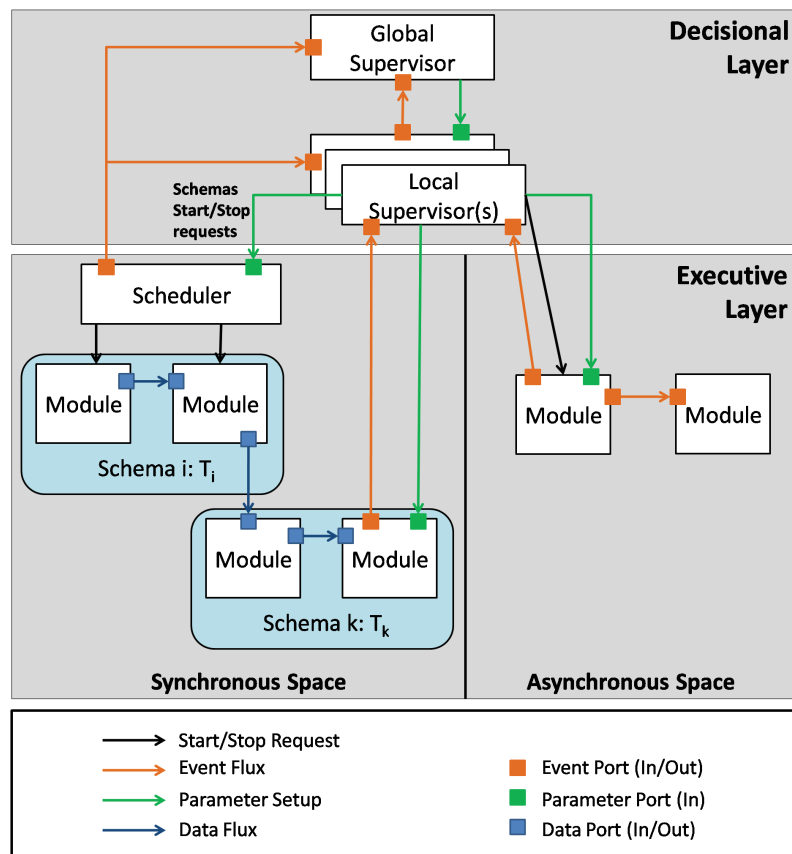


FIGURE 3.9 – Schéma de décomposition logicielle produite par ContrACT (figure tirée de [Pas10])

réception des données (e.g. données des encodeurs, ultrasons, bumpers). Le module **P3D** s'exécute dans un schéma de période 100 ms.

En plus, le module **P3D** remet en forme les données à envoyer et reçues du robot. Il interprète aussi les données des encodeurs pour déduire la pose du robot et ses vitesses linéaire et angulaire. La pose du robot peut être corrigée s'il y a une donnée de localisation provenant d'un module de localisation.

b. Module KIN : Gestion de la kinect

Le module périodique **KIN** intègre le driver du capteur kinect. Il permet de communiquer avec elle et gérer ses paramètres tout en acquérant les images RGB et/ou de profondeur. Notre utilisation de la kinect dans le cadre de la thèse est limitée à la récupération du flux vidéo.

Les images acquises sont exploitées par des algorithmes de localisation et de détection de forme. L'algorithme de localisation détecte des QR-codes présents dans l'image. Il dispose d'une connaissance sur les positions absolues de ces QR-codes dans l'environnement

et déduit ensuite la position du robot. Un algorithme de détection de forme peut aussi être exploité dans des applications d'inspection (e.g. repérer l'état d'une vanne).

Cependant, le flux de données transféré entre les modules est limité. C'est pourquoi, pratiquement l'ensemble de ces fonctionnalités (gestion du capteur kinect, localisation et détection de forme) est implanté au sein d'un seul module. L'exécution d'un des algorithmes peut alors être activée par l'envoi d'un paramètre au module. Nous distinguerons dans la suite le module **KIN** pour le driver, **LKI** pour la localisation basée sur la reconnaissance des QR-codes et **VKI** pour la détection de forme.

c. Module TLM : Gestion du télémètre laser

Le module **TLM** permet la communication de l'architecture avec le télémètre laser présent sur le robot (dirigé vers l'avant). C'est un module périodique récupérant et publiant les données lasers dans le repère du laser. Ce module possède un port de sortie de données, contenant toutes les valeurs laser (distances) datées. Un *thread* permet de lire les informations du télémètre sans bloquer le processus principal qui publie les données laser de façon synchrone.

Une duplication de ce module est faite pour gérer le deuxième télémètre laser (dirigé vers l'arrière). On l'appelle **TL2**.

d. Module LZA : Localisation

LZA est une méthode de localisation basée sur les données lasers (57 points) dans un environnement structuré et connu *a priori*. C'est une méthode de Chromo-Localisation qui permet d'estimer la pose réelle du robot⁵ (voir Annexe A). Une phase d'apprentissage hors-ligne est initialement faite (Figure 3.10). Un robot simulé se déplace linéairement avec un pas prédéfini dans l'environnement précisément modélisé. Ce pas représente la précision de la position estimée par l'algorithme et le nombre de points représente la précision de l'angle du robot (360/nombre de points). A chaque pas, les données lasers (simulées) sont stockées et classées selon leur énergie (somme des distances lasers divisé par le nombre total de faisceaux laser) et pourcentage (le nombre de données laser dépassant un seuil divisée par le nombre total de faisceaux laser) formant une signature caractéristique de la localisation du robot et de son cap.

La phase de localisation en-ligne, consiste à explorer les listes de données classées à

5. René Zapata, Lei Zhang and Lotfi Jaiem, "Stochastic Chromo-Localization LZA : a novel algorithm for mobile robot localization", submitted to ROBOTICA.

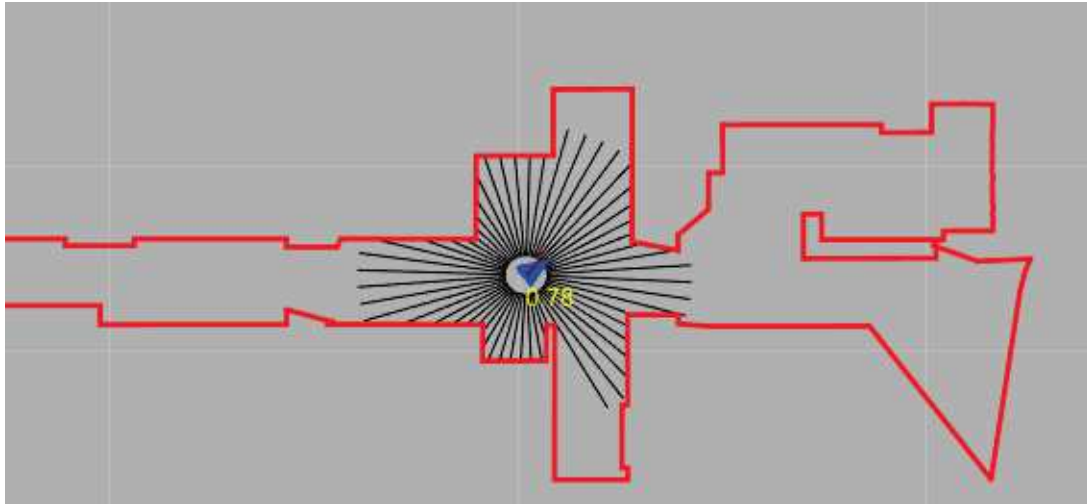


FIGURE 3.10 – Phase d'apprentissage de la méthode de Chromo-Localisation LZA

chaque période afin de trouver celle qui correspond le mieux aux données laser courantes. Une fois la position identifiée, la corrélation entre les deux signaux permet d'estimer l'angle réel du robot (Figure 3.11).

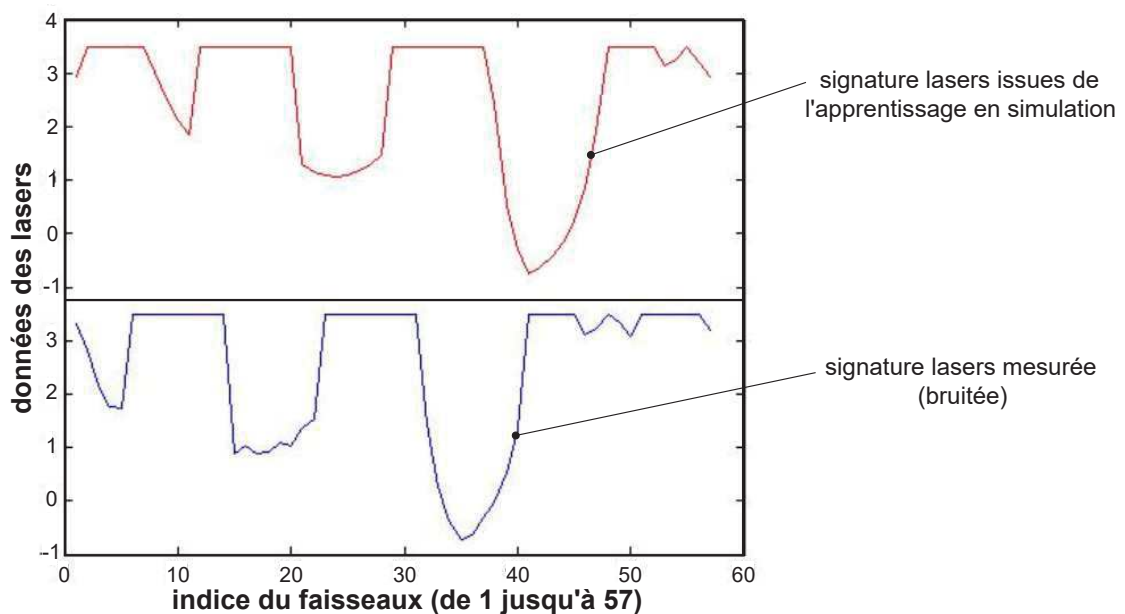


FIGURE 3.11 – Illustration des signatures laser issues de l'apprentissage et de la mesure

L'utilisation de cet algorithme nécessite l'emploi des deux lasers pour récupérer les 57 échantillons à 360° autour du robot. Pour cela, un module **FZL** assure la fusion des données provenant de **TLM** et **TL2** et poste en sortie les 57 données des télémètres uniformément réparties sur 360° autour du robot.

e. Module PAF : suivi de trajectoire SMZ

La méthode SMZ (Safe Maneuvering Zone) de suivi de chemin (PAtch Following) avec évitement d'obstacle a été développée dans [LZ12] (module périodique). Elle est composée de deux fonctions : le suivi de chemin avec garantie de convergence et l'évitement d'obstacle par déformation de trajectoire (Figure 3.12).

Le robot va d'abord s'approcher du chemin puis suivre un lièvre virtuel (triangle rouge), qui se déplace sur le chemin. Une loi cinématique assure la convergence du robot et le conduit asymptotiquement sur son chemin.

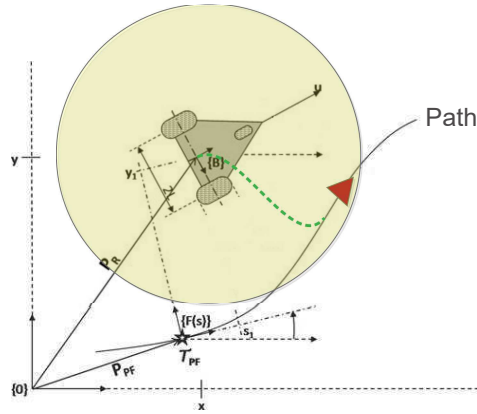


FIGURE 3.12 – Illustration de la fonction de suivi de chemin de la méthode **SMZ** [LZ12]

La fonction d'évitement d'obstacle s'enclenche si l'impact le plus proche détecté par les capteurs proximétriques se trouve dans la zone de sécurité autour du robot (cercle jaune). Dans ce cas, un nouveau lièvre pour l'évitement d'obstacle (triangle vert) remplace le lièvre de suivi de chemin (Figure 3.13).

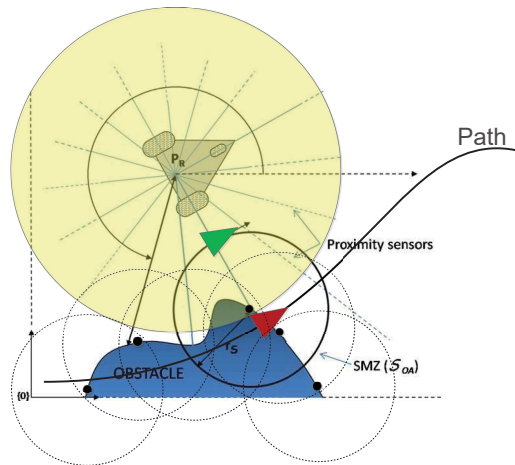


FIGURE 3.13 – Illustration de la fonction d'évitement d'obstacle de la méthode **SMZ** [LZ12]

Le lièvre d'évitement d'obstacle se positionne sur un cercle autour de l'impact le plus proche de rayon prédéfini. Ainsi la trajectoire est modifiée et le robot évite l'obstacle présent sur son chemin. Le robot rejoint par la suite le chemin initial en suivant de nouveau le lièvre rouge.

f. Module COC : Centrage

L'algorithme de centrage implémenté dans le module périodique **COC** (COrridor Centering) permet au robot de générer un chemin centré dans un environnement assimilable à un couloir. Le déplacement est assuré par le module **PAF**.

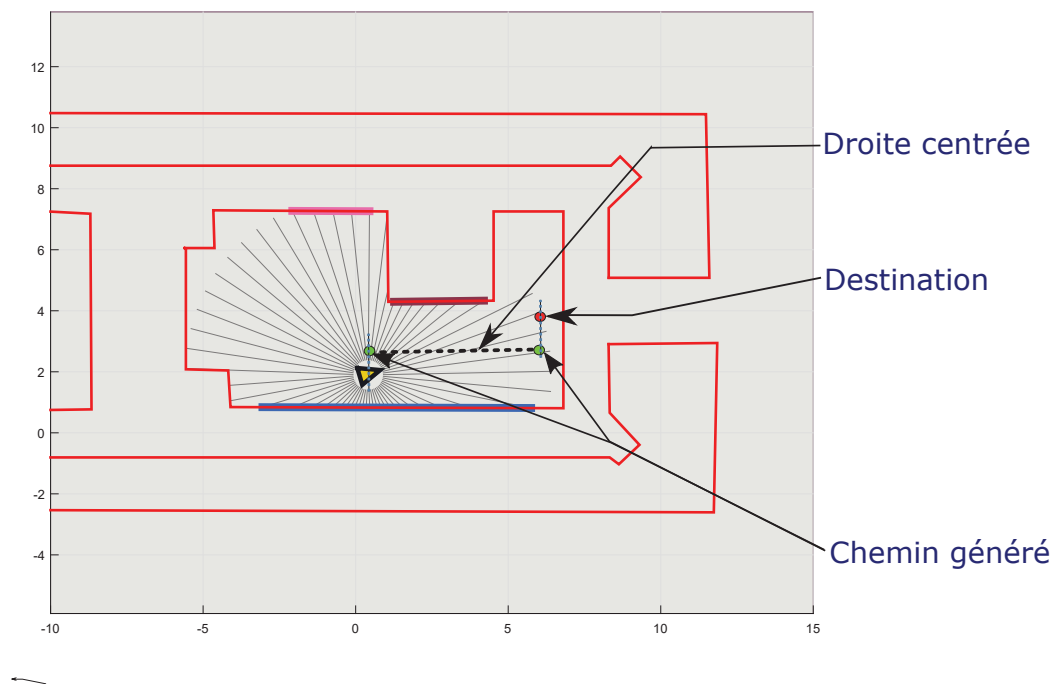


FIGURE 3.14 – Illustration de la détection de murs et génération de chemin de la méthode COC

L'algorithme de centrage se fait en trois étapes (Figure 3.14) :

- **Détection de couloir** : Le module reçoit en entrée les données laser provenant du module **FZL** (utilisé aussi par **LZA**). Une recherche récursive repère les éventuelles droites autour du robot. Parmi celles-ci, les droites parallèles sont identifiées. Il est considéré alors comme un couloir dans lequel il faut se centrer, le couple de droites parallèles les plus proches.
- **Génération de chemin** : Une fois le couloir détecté, un chemin est généré entre un point de départ et d'arrivée. Le point de départ représente la projection de la position du robot sur la droite centrée du couloir. Le deuxième point est la projection du

point destination sur la même droite. Le chemin est maintenant envoyé au module assurant le suivi de chemin.

- **Actualisation du chemin** : Au cours de son déplacement, l'algorithme continue son exécution et sa recherche de couloirs. S'il en existe, l'algorithme vérifie s'il s'agit du même couloir. Si un nouveau couloir est identifié (ou si le couloir actuel change de largeur), un nouveau chemin est généré et envoyé au module **PAF**.

g. Module **TF2** : Tourner vers

Le module périodique **TF2** (Turn Face To) permet de faire une rotation du robot sur lui même pour se diriger vers une cible. La position de cette cible est un paramètre de ce module et en fonction de l'angle et de la position courante, le robot déduit l'angle et le sens de rotation. La vitesse angulaire est aussi renseignée. Le module **TF2** envoie sur les ports de sortie les commandes des deux roues gauche et droite qui seront appliquées par le module **P3D**.

h. Modules **LAS**, **LA2** et **UST**

Ce sont des modules périodiques de changement de repère des données capteurs (**LAS** et **LA2** pour les laser et **UST** pour les sonars) au centre de rotation du robot (point milieu de l'essieu des deux roues). Ils reçoivent respectivement les données provenant des modules **TLM**, **TL2** et **P3D** dans les repères des capteurs. Ce changement de repère dépend de la position physique des télémètres ainsi que de la répartition et de l'orientation des sonars sur le robot. Les flux de sorties de ces modules contiennent les distances corrigées, la date d'acquisition ainsi que les angles des différents points capteurs.

i. Modules **FAS** et **DIP**

Un module **FAS** (Fusion All Sensors) fusionne périodiquement les données produites par **LAS**, **LA2** et **UST**. Il peut fusionner les données de tout ou partie de ces modules ou juste transmettre les données d'un seul capteurs selon le paramétrage choisi.

Le module **DIP** (Détection Impact Proche) permet d'identifier parmi les données fusionnées par **FAS** la distance et l'angle par rapport au robot de l'impact le plus proche. Cette information peut être par la suite produite exploitée, entre autres, par le module de contrôle du robot **PAF**.

j. Modules **WTT** et **CDR** : Gestion des périphériques

Le module périodique **WTT** gère la communication avec le wattmètre embarqué sur le robot. Il permet de remettre à zéro la mesure de consommation et de modifier un certain nombre de réglages (e.g. résolution de mesure, sauvegarde) ainsi que la lecture de la puissance, courant, tension, temps écoulé depuis la remise à zero, etc.

Le module périodique **CDR** (CarD Relay) assure la communication avec les deux cartes de commande (USB-relay2) et il commande l'activation des différents relais. Ce module est réveillé suite à un événement de paramétrisation. Les paramètres sont le numéro du relais à commander, état (actif/non actif) du relais 1 et celui du relais 2.

k. Modules **OMP** : Suivi mission

Le module périodique **OMP** (Observ Mission Progress) assure le suivi de l'avancement de la mission tout en observant les marges de performance et la disponibilité des différentes ressources. Il met en œuvre d'une part la phase en ligne de l'approche de gestion des ressources. Il reçoit comme paramètres le plan de la mission et l'activité courante, et d'autre part la consommation énergétique ainsi que la position du robot provenant des modules **WTT** et **P3D**.

Le module **OMP** déduit la position linéaire du robot et réactualise les valeurs des marges d'énergie et de temps. Si la mission est estimée non réalisable suite à une marge négative ou si une ressource utilisée ou prévue d'être utilisée tombe en panne, le module envoie un événement en indiquant la coordonnée linéaire du robot (progression de la mission) et l'identifiant du module fautif s'il y en a.

l. Module **MPM** : Gestion des ressources guidée par la performance

Le module aperiodique **MPM** (Mission Performance Management) est le module le plus important dans le cadre de cette thèse. Il intègre la mise en œuvre de la majorité de l'approche. Il est exécuté avant la mission pour décider si elle est réalisable. Et si elle l'est, avec quelle affectation de ressources. Il dispose d'une connaissance a priori de l'environnement sous forme de carte composée de zones sous contraintes comme le montre le diagramme UML de la figure [3.15](#).

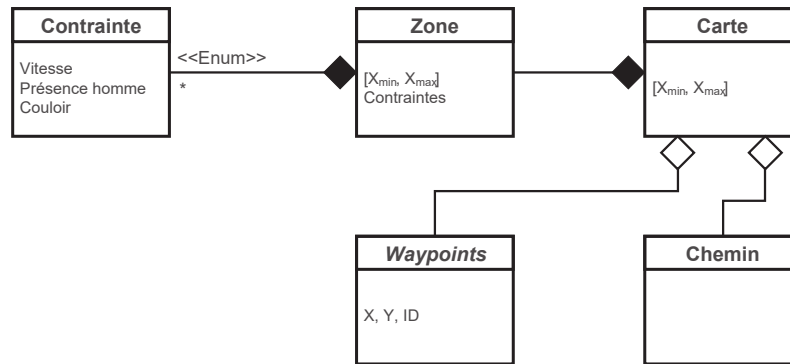


FIGURE 3.15 – Diagramme UML de la carte d’environnement

Le module **MPM** reçoit une description de la mission à réaliser sous la forme d’un fichier XML à partir duquel la mission est modélisée de point de vue utilisateur (Figure 3.16). Cette description comporte la séquence d’objectifs à réaliser, chacun composé d’un ensemble de tâches. Les ressources permettant de réaliser une tâche sont également identifiées.

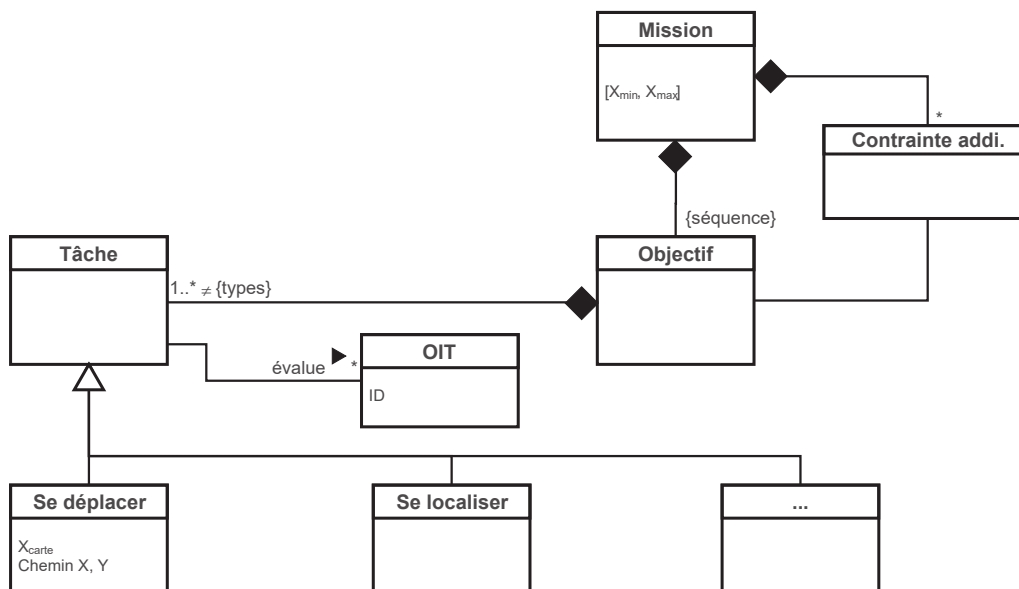


FIGURE 3.16 – Diagramme UML de la mission à réaliser : point de vue utilisateur

Un Scénario de Mission Détaillé *SMD* (voir section 2.2.2.2) est par la suite établi à partir de la description précédente de la mission et des différentes contraintes présentes (Phase P1) (Figure 3.17). La phase d’estimation hors ligne de la performance peut alors commencer pour définir une *SAR* si elle existe (voir Définition 9). Une *AI* est choisie et les schémas correspondants (ensemble de modules) sont activés et la mission commence alors.

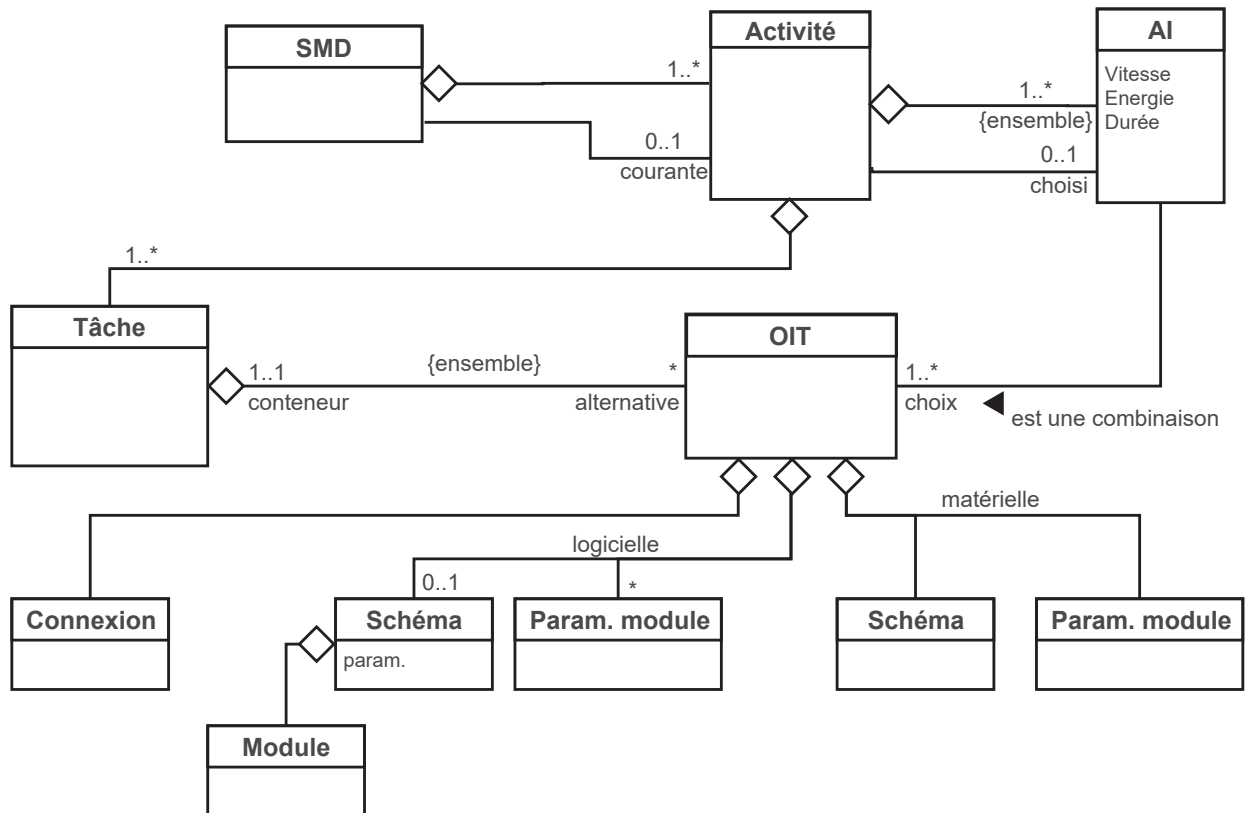


FIGURE 3.17 – Diagramme UML du scénario de mission détaillé

Conformément à la phase P3 de l'approche proposée, ce module est à nouveau exécuté suite à un événement reçu du module OMP indiquant une impossibilité de réalisation de la mission pour une dérive de performance ou une faute sur une ressource utilisée. La nouvelle *SAR* est alors calculée, si elle existe, en fonction du Reste du Scénario de Mission Détaillé *RSMD* (voir section 2.2.4).

3.1.2.3 Option d'Implémentation de Tâches OIT retenues

La redondance matérielle et logicielle offerte par notre système robotique permet d'avoir plusieurs OIT pour réaliser une tâche. Cependant, pour simplifier le cas d'étude sur lequel nous allons détailler la méthodologie proposée. Le nombre d'OIT par tâche a été volontairement limité. Le tableau suivant récapitule ces dernières.

Tableau 3.1 – Liste des OIT par tâche

Tâche	Déplacement Vers DV		LocalisationN LN		AnalysE d’ImaGe AEIG		Se diriger Vers SV	
OIT/Description (capteurs)	$S_{_LUS}$	Suivi de Chemin SMZ ($laser_1$ et sonars)	LOD	Localisation par ODométrie	DV	Détection de Vanne	TV	Tourner Vers une cible
	$S_{_US}$	Suivi de chemin SMZ (sonars)	LQR	Localisation par QR-codes (Kinect)				
	$S_{_L}$	Suivi de chemin SMZ ($laser_1$)	LCH	Localisation en utilisant la méthode de CHromo-Localisation (2 lasers)				
	$S_{_2LUS}$	Suivi de chemin SMZ (2 lasers et sonars)						
	$S_{_2L}$	Suivi de chemin SMZ (2 lasers)						
	$C_{_2L}$	Centrage (2 lasers)						
	$C_{_2LUS}$	Centrage (2 lasers et sonars)						

3.2 Mission de patrouille (*patrolling*)

Après avoir présenté l'architecture matérielle et logicielle du système robotique cible qui sera utilisé pour valider l'approche proposée, nous allons présenter dans cette section la mission de patrouille qu'il va réaliser. Nous allons tout d'abord décrire les objectifs de la mission d'inspection puis les contraintes environnementales et de performance imposées.

3.2.1 Description de la mission

La mission de patrouille visée est une reproduction d'une mission d'inspection dans un contexte industriel. La salle d'expérimentation située au sous sol du bâtiment 5 du LIRMM constitue son point de départ **D**. Deux vannes V_1 et V_2 situées au niveau des deux couloirs CR_1 et CR_2 (Figure 3.18) doivent être inspectées.

Le robot doit alors partir du point de départ **D** pour atteindre V_1 et V_2 et inspecter leurs états (ouvert / fermé - Figure 3.19). Il faut noter que la position des vannes est connue. Le robot revient par la suite à son point de départ.

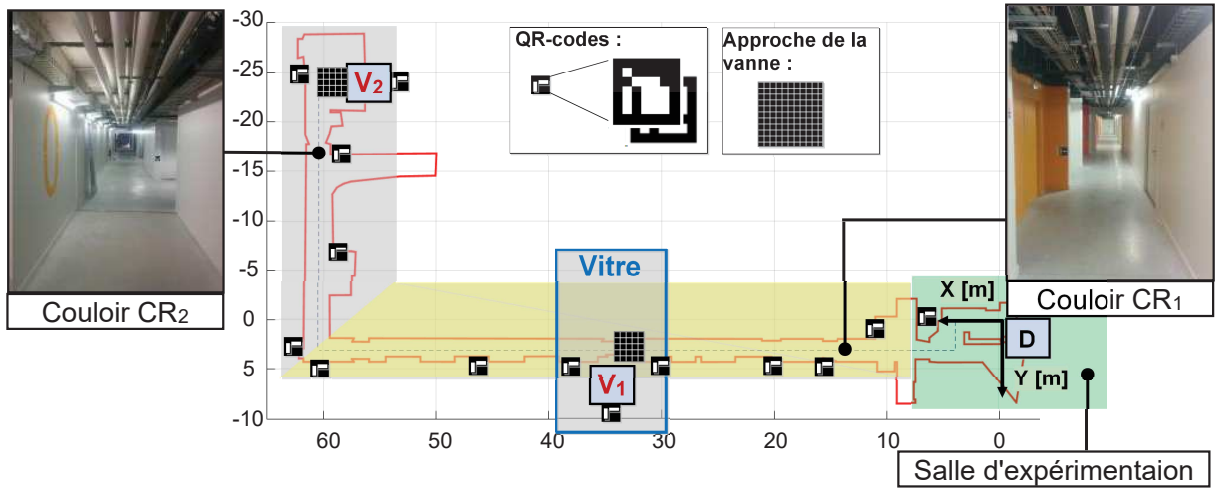


FIGURE 3.18 – Description de l'environnement de la mission

Conformément à la définition d'une mission robotique dans la section 2.2.1.1, la mission se compose d'une séquence d'objectifs récapitulés dans le tableau 3.2. Les objectifs de déplacement O_I , O_V et O_{IX} font appel à deux tâches robotiques (locomotion DV et localisation LN dont les OIT correspondantes ont été détaillées dans le tableau 3.1). Les objectifs O_{II} , O_{IV} , O_{VI} et O_{VIII} permettant de se diriger vers les vannes, ou vers la suite du chemin, se réalisent aussi avec deux tâches (rotation TV et localisation LN). Enfin, les objectifs d'inspection de l'état de la vanne O_{III} et O_{VII} se font avec une unique tâche d'analyse d'image AEIG. Lors de son aller retour le robot parcourt environ 190 m.



FIGURE 3.19 – Vannes à inspecter

Tableau 3.2 – Description de la séquence d’objectifs de la mission cible

	Description des objectifs	Tâches
O_I	Aller à V_1	DV / LN
O_{II}	Se diriger vers V1	TV / LN
O_{III}	Inspecter l’état de V1	AEIG
O_{IV}	Se retourner	TV / LN
O_V	Aller à V2	DV / LN
O_{VI}	Se diriger vers V2	TV / LN
O_{VII}	Inspecter l’état de V2	AEIG
O_{VIII}	Se retourner	TV / LN
O_{IX}	Revenir au point de départ	DV / LN

La mission visée est maintenant définie (de point de vue utilisateur). L’environnement de cette mission ainsi que les contraintes relatives à ce milieu vont maintenant être présentées.

3.2.2 Environnement

L’aile du sous-sol dans lequel la mission va être réalisée se compose de la salle de manipulation (point de départ D), d’un premier couloir à la sortie de cette salle d’une longueur de 50 m (CR_1) et d’un deuxième couloir (CR_2) de 25 m (Figure 3.18).

Cet espace de travail a été instrumenté par des QR-codes utilisés pour la localisation en utilisant la kinect. Les positions ainsi que les orientations de ces balises sont connues au préalable par le robot. Un modèle de cet étage a été établi pour mettre en œuvre l’algorithme de localisation LZA présenté précédemment.

Certaines contraintes ont été considérées (Figure 3.18) :

- c_1 : La salle d'expérimentation et le couloir CR_1 permettent une présence humaine.
- c_2 : Le couloir CR_2 est sans présence humaine.
- c_3 : Une zone de vitre est dans le couloir CR_1 .
- c_4 : L'approche des vannes doit se faire avec une localisation précise basée sur les QR-codes.
- c_5 : Le robot ne doit pas dépasser la vitesses $v_{QR-codes}$ pour une localisation avec les QR-codes.

3.2.3 Contraintes de performance

Les contraintes de performance considérées pour appliquer l'approche de gestion des ressources dans la suite sont liées aux axes sécurité, énergie et durée. Les contraintes de performance suivantes sont définies :

- **Axe durée** : La contrainte de durée est fixée à $Perf_{C, \text{Durée}}$. L'exécution de la mission ne doit pas dépasser cette valeur.

$$\boxed{Perf_{SR, \text{Durée}} \leq Perf_{C, \text{Durée}}} \quad (3.1)$$

- **Axe énergie** : Étant donné que le système robotique dispose de deux batteries distinctes, deux indicateurs de performance sont définis pour l'axe énergétique :

- Consommation maximale autorisée sur la batterie du robot est $Perf_{C, \text{Énergie}}^{\text{Robot}}$.

$$\boxed{Perf_{SR, \text{Énergie}}^{\text{Robot}} \leq Perf_{C, \text{Énergie}}^{\text{Robot}}} \quad (3.2)$$

- Consommation maximale autorisée sur la batterie de l'ordinateur portable est $Perf_{C, \text{Énergie}}^{PC}$.

$$\boxed{Perf_{SR, \text{Énergie}}^{PC} \leq Perf_{C, \text{Énergie}}^{PC}} \quad (3.3)$$

- **Axe sécurité** : La prise en compte de l'axe sécuritaire se fait selon deux indicateurs :
 - Assurer l'évitement d'obstacles EO.

$$\boxed{Perf_{C, \text{Sécurité}}^{EO} = \text{Vrai}} \quad (3.4)$$

- Assurer l'innocuité In : Le robot ne doit participer à un éventuel choc avec une énergie maximum.

$$\boxed{Perf_{C, Securite}^{In} = Vrai} \quad (3.5)$$

La mise en œuvre des précédents axes de performance va être présentée dans le chapitre suivant. Cependant, la mission ne sera réalisable que si on prend en considération les axes de localisation et de stabilité. On ne détaillera pas dans cette thèse ces deux points de vue mais on les prendra en compte heuristiquement par :

- **Axe stabilité** : La fréquence d'exécution du contrôle fixée à 10 Hz assure expérimentalement la stabilité du robot.
- **Axe localisation** : La kinect assure la meilleure localisation et elle doit être utilisée à l'approche des vannes.

3.3 Conclusion

Le système robotique composé d'une plateforme mobile enrichie par des capteurs supplémentaires et d'un ordinateur embarqué, a été présenté dans ce chapitre. Il permet de réaliser la mission de patrouille avec une redondance matérielle et logicielle lui conférant un certain niveau de robustesse face aux différentes situations adverses pouvant perturber sa réalisation.

Le chapitre suivant présente la mise en œuvre des deux phases statiques de l'approche de gestion des ressources proposée, basées principalement d'une part sur l'identification des modèles d'estimation de la performance selon les axes sécurité, énergie et durée et d'autre part sur la construction d'un scénario de mission détaillé *SMD* et d'une solution d'allocation de ressources *SAR*.

Chapitre 4

Mise en œuvre de la phase statique de la méthodologie d'affectation des ressources

Sommaire

4.1	Méthodologie : Phase préliminaire P1	106
4.1.1	Axe sécurité	106
4.1.2	Axe énergie	116
4.1.3	Axe durée	138
4.1.4	Axe localisation	142
4.1.5	Axe stabilité	143
4.1.6	Identification du Scénario de Mission Détaillé (SMD)	145
4.1.7	Conclusion	153
4.2	Méthodologie : Phase hors ligne P2	153
4.2.1	Mise en œuvre de la phase P2 sur SMD sans performances booléennes et avec prise en compte de l'axe durée	154
4.2.2	Mise en œuvre de la phase P2 sur SMD avec prise en compte de l'axe sécurité (booléen) et de l'axe durée (continu)	155
4.2.3	Mise en œuvre de la phase P2 sur SMD avec prise en compte des axes booléens et continus	157
4.3	Conclusion	158

4.1 Méthodologie : Phase préliminaire P1

Nous avons présenté dans les chapitres précédents l'approche de gestion des ressources ainsi que le système robotique et la mission cible. Dans cette partie, la première phase de la méthodologie va être appliquée sur ce cas d'étude. Nous abordons dans un premier temps, l'étape d'identification des indicateurs, inducteurs et modèles d'estimation de performance en commençant par les axes principaux. Dans un second temps, nous détaillons les *SMD* relatifs à la mission et les contraintes de performance et d'environnement pour les différents points de vue.

4.1.1 Axe sécurité

Comme indiqué dans les sections 2.2.2.1 et 1.2.1.6, l'identification des éléments impactant la performance se fait en utilisant l'outil graphique "causes à effets". Il permet de construire un graphe structurant les inducteurs et indicateurs de performance ainsi que leurs relations.

4.1.1.1 Graphe des indicateurs/inducteurs de performance de l'axe sécurité

En premier lieu le diagramme d'Ishikawa [IL85] de l'axe sécurité illustré dans la figure 4.1 est construit. Il dévoile les éléments provenant de différents centres d'intérêt (5 M : milieu (environnement), méthode, matière, matériel et main d'œuvre (opérateur)) et qui impactent ce point de vue de la performance. On peut remarquer que l'opérateur n'a pas d'influence sur la sécurité puisqu'on est dans le cas d'une mission autonome.

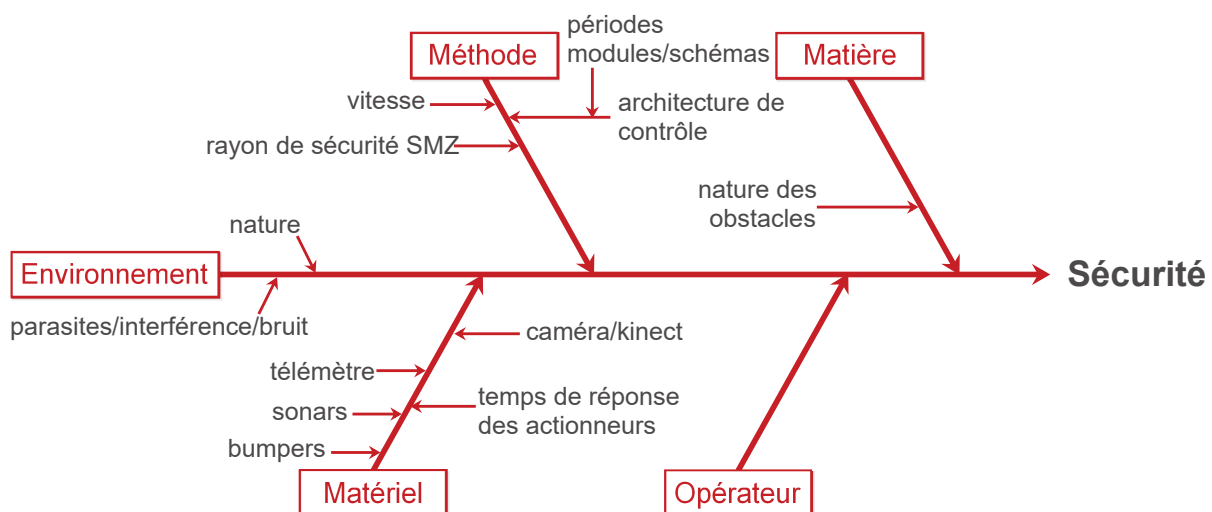


FIGURE 4.1 – Diagramme causes-effets pour l'axe sécurité appliqué au système d'étude

Le diagramme d'interdépendance de la figure 4.2 montre les relations entre les inducteurs et les indicateurs de performance (nœuds). Deux nœuds sont principalement concernées (les flèches sont uniquement entrantes). Ce sont les indicateurs de performance sécuritaire (rectangles verts) :

- **Capacité d'évitement d'obstacle** : Pour assurer l'évitement d'obstacles statiques, le robot doit être en mesure de **percevoir** son environnement et de **réagir** si nécessaire.
 - La perception dépend principalement des capteurs embarqués ainsi que de leurs caractéristiques (précision, fréquence et nature des objets visibles). Il faut remarquer que la disposition de ces capteurs (hauteur et orientation) influe la capacité de perception des objets de géométries et de tailles variées. De plus, les données de ces capteurs sont acquises et traitées au sein d'une architecture de contrôle exécutant des schémas à différentes fréquences. D'où l'impact de la dimension architecturale sur la (les) durée(s) de traitement des données. La capacité de perception dépend aussi de la nature des objets perçus et de la technologie des capteurs utilisées (e.g. les vitres ne sont pas détectées par les lasers).
 - La capacité d'évitement d'obstacles est aussi impacté par la capacité de réaction du système robotisé. Cette dernière dépend des actionneurs qui, en fonction de leurs caractéristiques, ont un temps de réponse lié à la dynamique du robot (masse et vitesse). Elles dépend aussi de l'algorithme qui assure l'évitement d'obstacle.
- **Innocuité** : L'indicateur de performance sécuritaire d'innocuité indique qu'en cas d'impact avec un obstacle extérieur à cause de son dynamisme ou suite à une défaillance matérielle ou logicielle, le robot ne peut apporter une énergie cinétique supérieure aux limites fixées par les réglementations et les normes en cours. Cette énergie dépend bien évidemment de la vitesse et de la masse du robot.

Les nœuds en jaune dans la figure 4.2 représentent les inducteurs sur lesquels il est possible d'agir pour piloter la performance. Ils correspondent soit à des ressources matérielles telles que les capteurs, soit à des ressources logicielles ainsi qu'à des paramètres tels que la vitesse du robot, le rayon de sécurité, les gains de la lois de commande, etc. Le système robotique cible ne présente pas de redondance d'actionnement. Si c'était le cas, le choix d'un actionneur parmi ceux disponibles aurait été un inducteur de performance supplémentaire. De même, la masse du robot est fixe pour la mission visée. On peut imaginer une mission où le robot doit livrer des colis de différentes masses. Dans ce cas, la masse du robot serait un inducteur de performance possible impactant à la fois les indicateurs d'innocuité et d'évitement d'obstacles (via la capacité de réaction).

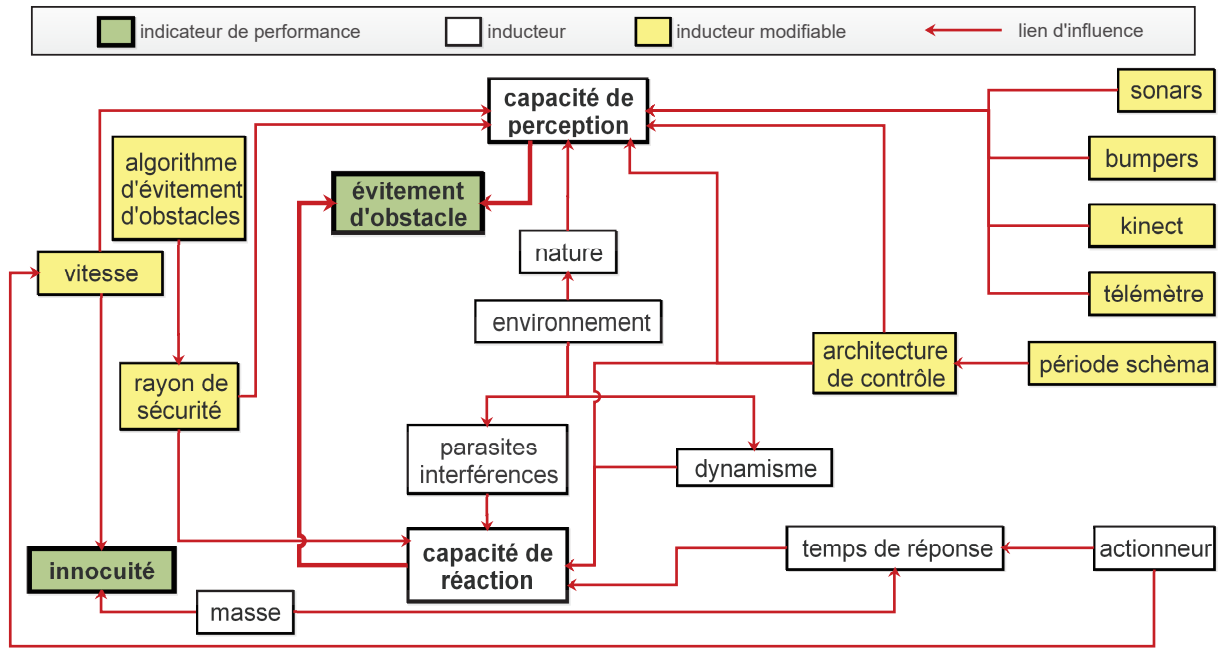


FIGURE 4.2 – Diagramme d'interdépendance de l'axe sécurité appliqué au système d'étude

On retrouve la même vision de la sécurité dans [IIN03] où les auteurs différencient entre la phase précédant la collision et la collision avec impact (Table 4.1).

Tableau 4.1 – Classification des stratégies de sécurité [IIN03]

		stratégie de contrôle	stratégie de conception
avant la collision	éviter la collision	distance	-
	minimiser la force de l'impact	vitesse	-
		moment d'inertie	masse
après la collision	atténuation diffusion	raideur	coque compliance des articulations
			surface forme

Passons maintenant à l'étape d'estimation des performances suivant les indicateurs de performance sécuritaire révélés par l'étude précédente.

4.1.1.2 Modèles d'estimation de la performance sécuritaire

La question sécuritaire est prégnante en robotique. La norme ISO 10218-1 [ISO11b] encadre les exigences de sécurité des robots industriels dans le cadre collaboratif. Cependant les règles énoncées ne peuvent encadrer le déploiement des robots de services évoluant

dans des contextes très différents à ceux industriels. La norme ISO/DIS 13482 [ISO11a] s'intéresse plus particulièrement aux dispositifs robotiques non médicaux pour les soins personnels c'est-à-dire aux robots thérapeutiques et d'assistance personnelle pour les personnes en perte d'autonomie. Un autre groupe de travail, le TC 184/DC 2/WG 8 porte plus largement sur la robotique de service. Encore en retard par rapport au monde industriel, les normes dans le cadre de la robotique de service en présence humaine restent insuffisantes et nécessitent plus de formalisation.

a. Évitement d'obstacles

Dans ce cadre nous allons nous intéresser plus particulièrement au déplacement dans le plan d'un robot mobile autonome et à sa capacité d'évitement d'obstacles. Comme le montre la figure 4.2, l'aspect sécuritaire de l'évitement d'obstacle ne peut pas se limiter à la capacité de réaction. En effet pour éviter un obstacle il faut être capable de le détecter et donc disposer du capteur permettant de le faire. En fonction de la nature de l'environnement (couleur, type de revêtement, degré de transparence, etc.) on peut donc être amené à mobiliser différents dispositifs. Ceci est exprimé par une contrainte c_i exigeant l'utilisation ($\{ressource_i\}$) ou la non utilisation ($\{\overline{ressource_i}\}$) d'un ensemble de ressources logicielles et/ou matérielles. A titre d'exemple les US moins précis et rapides que le laser sont capables de détecter des vitres alors que ce dernier ne le peut.

Pour assurer la sécurité du déplacement il faut donc garantir que l'on dispose des capteurs adaptés à l'environnement si on connaît ce dernier, ou de l'ensemble des capteurs nécessaires à sa découverte lors d'une mission d'exploration. Dans le cas d'un univers connu, on peut donc imaginer disposer d'une cartographie de ce dernier délimitant un zonage précisant la nature des capteurs nécessaires.

Disposer des capteurs et actionneurs permettant de mettre en œuvre l'évitement d'obstacles ne garantit pas la réussite de cette manœuvre. Le robot doit engager un évitement d'obstacle si ce dernier est présent dans sa zone de sécurité de diamètre $d_{securite}$ (avec le principe de la SMZ précédemment présentée). En effet, la distance d'évitement d'obstacle d_{EO} équivalente à celle parcourue par le robot durant les phases de perception ($d_{perception}$ sans rafraichissement des données de capteurs proximétriques) et de réaction ($d_{reaction}$) doit être inférieure ou égale à celle le séparant de l'obstacle à éviter (d_{sec}) (Figure 4.3).

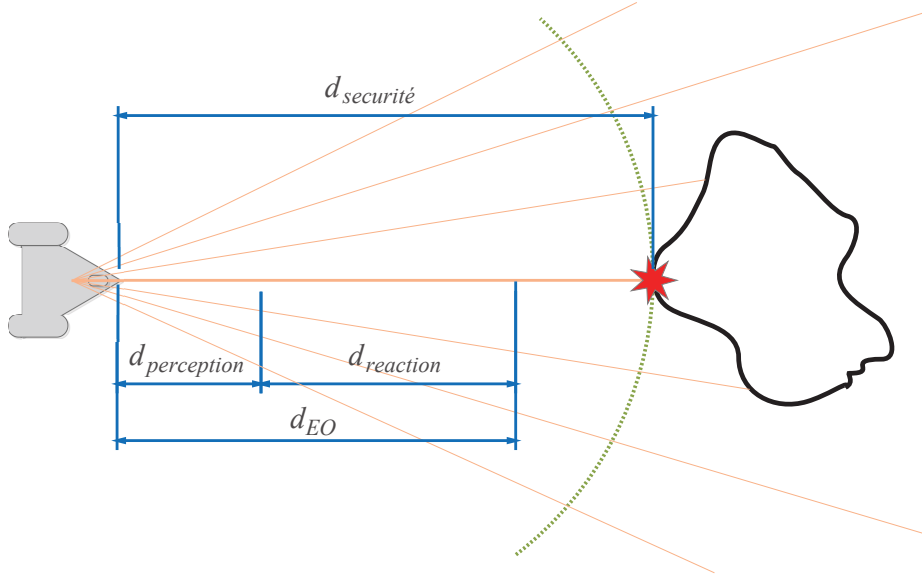


FIGURE 4.3 – Evitement d'obstacle : Distance de sécurité, de perception et de réaction

L'ensemble de ces conditions qui définissent le respect de l'axe sécuritaire suivant l'indicateur d'évitement d'obstacle OA sont définies par la règle suivante :

Règle 1:		
$Perf_{SR, Securite}^{EO} = \text{Vrai}$		
Si $d_{EO} = d_{perception} + d_{reaction} \leq d_{securite}$	et	
Si $\{ressource_i\}$ est utilisé	et	
Si $\{\overline{ressource_i}\}$ est non utilisé		

Afin de vérifier la première conditions de la règle 1, il faut maintenant identifier $d_{perception}$ et $d_{reaction}$:

- La distance $d_{perception}$ correspond à la distance parcourue entre deux acquisitions successives de données de perception. Le robot évolue en étant "aveugle" entre ces deux instants. Durant ce déplacement le robot garde des données non actualisées (Figure 4.4.a). Cette durée/distance dépend bien évidemment de la vitesse du robot v_r et de la fréquence de rafraichissement f_{raf} (4.1) des données capteurs.

$$d_{perception} = \frac{v_r}{f_{raf}} \quad (4.1)$$

Cette fréquence dépend de la fréquence d'acquisition du capteur f_{cap_i} et de la fréquence de l'architecture de contrôle $f_{sch_{capteur}}$ (schéma dans lequel le module driver du capteur est exécuté). Comme le montre l'équation 4.2 f_{raf} est un multiple de

l'inverse de l'entier supérieur du quotient $f_{sch_{capteur}}$ et f_{cap_i} . Ainsi si la fréquence du capteur est plus grande que celle du driver (Figure 4.4.b), ses données ne seront reçues qu'à chaque cycle d'exécution du schéma driver de l'architecture. D'autre part, si ce dernier est plus rapide que le capteur, ses données ne seront effectivement mises à jour qu'au prochain cycle du schéma driver (Figure 4.4.c).

$$f_{raf} = \frac{1}{\left\lceil \frac{f_{sch_{capteur}}}{f_{cap_i}} \right\rceil} \cdot f_{sch_{capteur}} \quad (4.2)$$

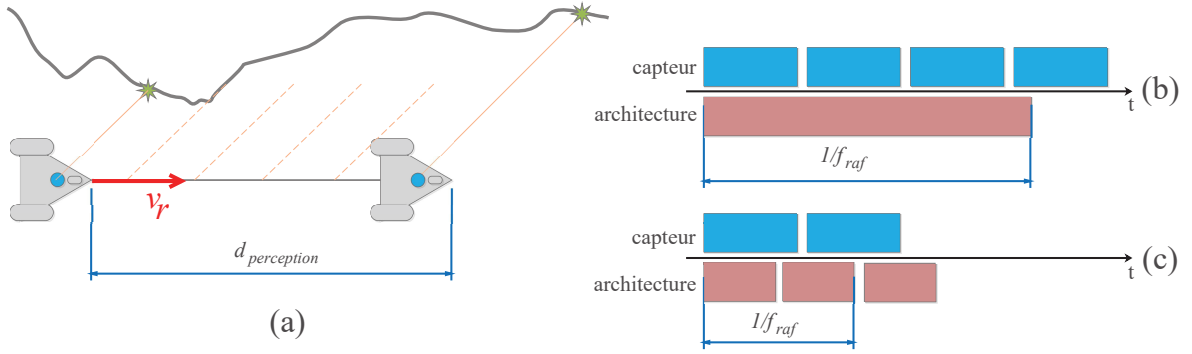


FIGURE 4.4 – Illustration de la distance parcourue lors de la phase de perception

- La distance de réaction $d_{reaction}$ correspond à la distance parcourue d_{TR} pendant le calcul de la commande à appliquer (distance de temps de réponse) à laquelle on ajoute la distance $d_{contournement}$ qui est celle parcourue en direction de l'impact durant la manœuvre de contournement de ce dernier (Figure 4.5) :
- Le temps de calcul d'une commande dépend de la fréquence du schéma de contrôle exécuté f_{cont} . La distance de temps de réponse d_{TR} peut être déduite en fonction de la vitesse du robot (4.3).

$$d_{TR} = \frac{v_r}{f_{cont}} \quad (4.3)$$

- Une fois la commande calculée, l'évitement d'obstacle dans le cas le plus critique (impact juste en face) se fait en saturant les actionneurs au maximum. Il faudra alors que l'obstacle détecté soit à une distance supérieure ou égale à la moitié de la largeur du robot l_{robot} à laquelle il faut ajouter une marge conservatrice ε pour pouvoir être évité. La méthode de suivi de chemin avec évitement d'obstacle utilisée dans le cadre de cette thèse est celle proposée dans [LZ12] où cette distance est appelée r_{SMZ} . Il reste maintenant la manœuvre de contournement qui dépend de la nature du robot. Un véhicule peut réaliser un virage avec un rayon de courbure minimal $1/C$ où C est la courbure de la

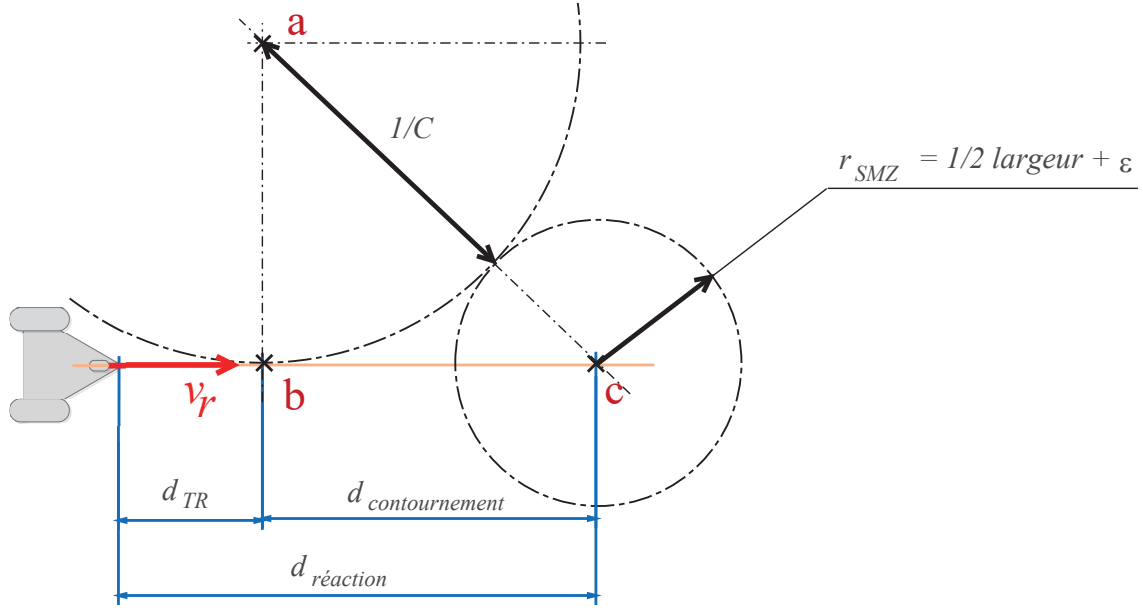


FIGURE 4.5 – Illustration de distance nécessaire pour contourner un obstacle

trajectoire. Une courbure maximale C_{max} est réalisable (4.4). Elle dépend de la vitesse angulaire maximale du robot r_{max} et la vitesse d'avance minimale correspondante v_{rmin} .

$$C = \frac{r}{v_r} \quad \text{et} \quad C_{max} = \frac{r_{max}}{v_{rmin}} \quad (4.4)$$

Il est maintenant possible de déduire la distance de contournement $d_{contournement}$ (4.5) en considérant le triangle rectangle abc de la figure 4.5.

$$d_{contournement} = \sqrt{2 \cdot \left(\frac{r_{SMZ}}{\frac{r}{v_r}} \right) + r_{SMZ}^2} \quad (4.5)$$

Comme le robot cible est de type unicycle, il est capable de tourner sur lui même avec une vitesse d'avance v_{rmin} nulle. La courbure C est alors infinie et le rayon de courbure est nul. La distance de contournement se réduit alors à r_{SMZ} .

La première condition de la règle 1, peut être explicitée comme suit :

$$\begin{aligned} \frac{v_r}{f_{raf}} + \frac{v_r}{f_{cont}} + \sqrt{2 \cdot \left(\frac{r_{SMZ}}{\frac{r}{v_r}} \right) + r_{SMZ}^2} &\leq d_{securite} \\ \frac{v_r}{f_{raf}} + \frac{v_r}{f_{cont}} + r_{SMZ} &\leq d_{securite} \\ \frac{v_r}{f_{raf}} + \frac{v_r}{f_{cont}} + \frac{1}{2} \cdot l_{robot} + \varepsilon &\leq d_{securite} \end{aligned} \quad (4.6)$$

Si l'équation (4.6) est vérifiée, l'évitement de tout obstacle statique visible par le capteur cap_i peut être garantie.

Considérons maintenant le cas du système robotique cible doté de 16 US ($f_{US} = 1.56$ Hz pour un scan complet) et des lasers ($f_{laser} = 10$ Hz). Les drivers de ces capteurs sont exécutés dans des schémas à une période de 10 Hz. La fréquence du schéma de contrôle f_{cont} est aussi fixée à 10 Hz. La largeur du Pioneer-3DX est 0.45 m et on pose $\varepsilon = 0.1$ m. En fonction de la vitesse du robot, on peut maintenant tracer la distance nécessaire pour éviter un obstacle et déduire la vitesse maximale du robot pour une r_{sec} donnée (Figure 4.6).

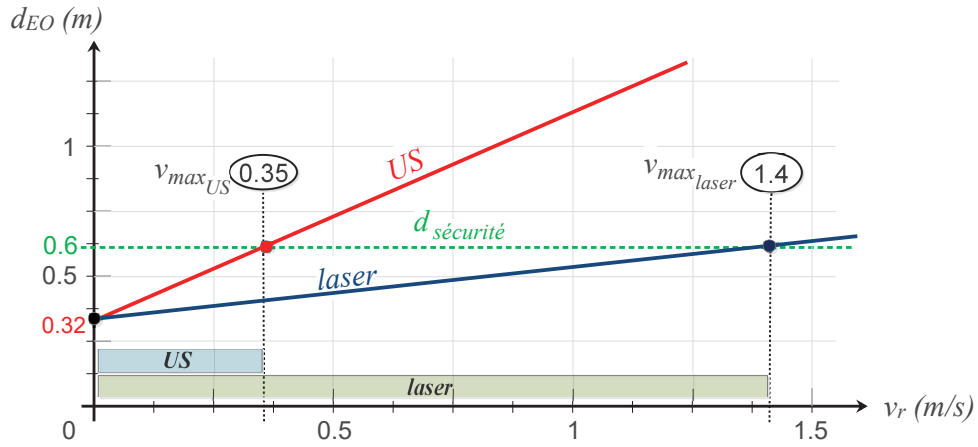


FIGURE 4.6 – Vitesses permettant d’assurer l’évitement d’obstacles et l’innocuité en fonction des capteurs

La figure 4.6 montre les plages de vitesses admissibles pour un exemple de $d_{securite}$ limitée à 0.6 m. Le laser ayant une fréquence d’acquisition nettement plus élevée que celle des sonars permet d’assurer un évitement d’obstacle statique à une vitesse (1.4 m/s) supérieure à celle des sonars (0.35 m/s). Évidemment si on veut anticiper l’évitement d’obstacle et apporter une réponse moins extrême que la saturation maximale des actionneurs il faut que d_{EO} soit nettement inférieure à d_{sec} . De toute façon lorsqu’un obstacle (dynamique et en déplacement) apparaît brusquement à une distance inférieure à d_{sec} , la collision ne

peut être évitée. Dans ce cas, assurer la sécurité, implique de rajouter aux contraintes sécuritaires d'évitement d'obstacle celles liées à l'innocuité traitées dans la suite.

b. Innocuité

Lorsqu'un obstacle, dynamique, apparaît brusquement à une distance inférieure à une distance critique, le robot ne pourra pas éviter la collision. [Pag16] cite 5 causes possibles d'accident avec un véhicule guidé automatique :

- mauvaise conception de l'environnement de travail.
- défaillance des systèmes de détection.
- insuffisance de la réaction suite à une présence humaine.
- autorisation de l'accès de personne dans les zones de circulation des véhicules.
- mauvaise mise en œuvre des règles opératoires.

En plus des causes mentionnées précédemment, s'ajoutent celles liées à la mauvaise conception du véhicule (bordures lisses) ou du site de travail (accessibilité possible dans des endroits interdits). Ainsi, si on veut garantir la sécurité du robot et de son environnement, ce type d'événement doit être pris en compte. [Mm12] définit l'innocuité en relation avec la sécurité comme l'absence de conséquences catastrophiques. L'innocuité est aussi définie dans [IEC99] comme une absence de risque inacceptable. Ces deux définitions montrent l'importance de la notion de *seuil* acceptable.

[Ala+06] affirme que le critère le plus important pour limiter les blessures en cas de collision est de limiter la masse de l'objet en mouvement. Cependant, dans l'édition de Techniques de l'ingénieur sur la sécurité en robotique [Pag16], une vitesse limite de 0.25 m/s est recommandée. Cette limite est donc indépendante de la masse et la dynamique du véhicule en déplacement. Dans le domaine de l'automobile, la NHTSA¹ définit en 1972 le critère *HIC*² (4.7) pour évaluer les risques. Dans cette equation t_1 et t_2 désignent les deux instants entre lesquels le critère *HIC* atteint son maximum et a est l'accélération mesurée au niveau de la tête d'un mannequin d'essai d'un *crash-test*.

$$HIC = \left\{ \left[\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} a(t) dt \right]^{2.5} (t_2 - t_1) \right\}_{max} \quad (4.7)$$

Cependant ce critère n'est pas suffisant pour évaluer les blessures qui peuvent être conséquentes même à de faibles accélérations. Pour cela, [HASH09] et [HASH08] ont

1. NHTSA : *National Highway Traffic Safety Administration*

2. HIC : *Head Injury Criteria*

proposé d'utiliser la possibilité de fracture du crâne comme un critère expérimental d'évaluation des blessures.

La problématique de choc avec un être humain a été formalisée par des normes françaises (décret 95-589 du 6 mai 1995) dans le domaine de l'*Airsoft*. Ces normes limitent l'énergie (*seuil*) d'un projectile de répliques d'armes. Elles imposent qu'un projectile ne peut dissiper une énergie supérieure à 2J (sans autorisation) et à 4J (avec déclaration en gendarmerie – d'une arme de 7^{ème} catégorie). On trouve par ailleurs la même énergie cinétique admissible en cas de choc de 4J dans la norme ISO [ISO/TC 184/SC 2] et [HJI06]. Le robot ne peut pas alors se déplacer au-delà d'une vitesse qui engendrerait un choc dangereux (entraînant une sensation de douleur) pour l'être humain heurté. Cela permettra de limiter l'énergie à dissiper.

Compte tenu de la masse m de notre robot d'environ 25 kg et de la plage de vitesses V possibles on obtient la courbe de la Figure 4.7 qui illustre (4.8).

$$E_{max_{choc}} = \frac{1}{2} \cdot m \cdot V_{choc}^2 \quad (4.8)$$

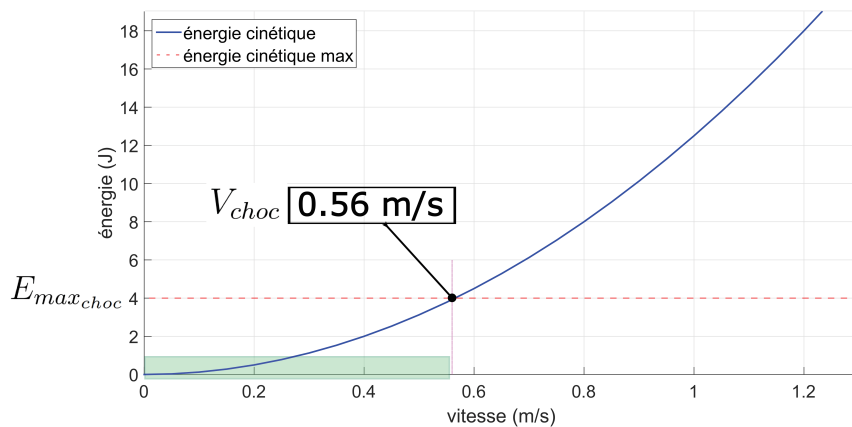


FIGURE 4.7 – Énergie cinétique versus vitesse pour un objet en mouvement de 25 kg

Étant donnée que la masse de notre robot est constante, cela conduit à une vitesse maximale de déplacement $V_{choc} = 0.56 \text{ m/s}$ pour que l'énergie $E_{max_{choc}}$ admissible reste limitée à 4J. Outre l'aspect dynamique, il ne faut pas négliger la conception matérielle du robot qui ne doit pas présenter de risque de cisaillement, accrochage, sectionnement, etc [HJI06].

Donc, dans une zone de présence humaine, la sécurité selon l'indicateur de performance d'innocuité In pour le robot étudié est fonction de la vitesse du robot v_r . Elle est évaluée par la règle suivante :

Règle 2:

$$Perf_{C, Securite}^{In} = \text{Vrai} \\ \text{Si } v_r \leq V_{choc}$$

4.1.2 Axe énergie

L'axe d'énergie est un point central pour la réalisation d'une mission robotique autonome. L'étude de cet axe va débiter, comme pour l'axe sécuritaire, par l'identification graphiques des éléments impactant cet axe suivi par l'établissement des modèles d'estimation de l'énergie.

4.1.2.1 Graphe des indicateurs/inducteurs de performance de l'axe énergie

Le diagramme d'Ishikawa de la figure 4.8 montre les principaux éléments qui impactent la consommation énergétique d'un système robotique. Comme pour la sécurité, et puisque le robot est autonome (après réception de la mission à réaliser), l'homme n'a aucune influence sur la consommation énergétique. Comme le montre cette figure, la consommation d'énergie dépend principalement des méthodes mises en œuvre (algorithme de commande, chemin planifié) et des ressources matérielles utilisées (capteurs et actionneurs) pour faire évoluer le robot dans son environnement.

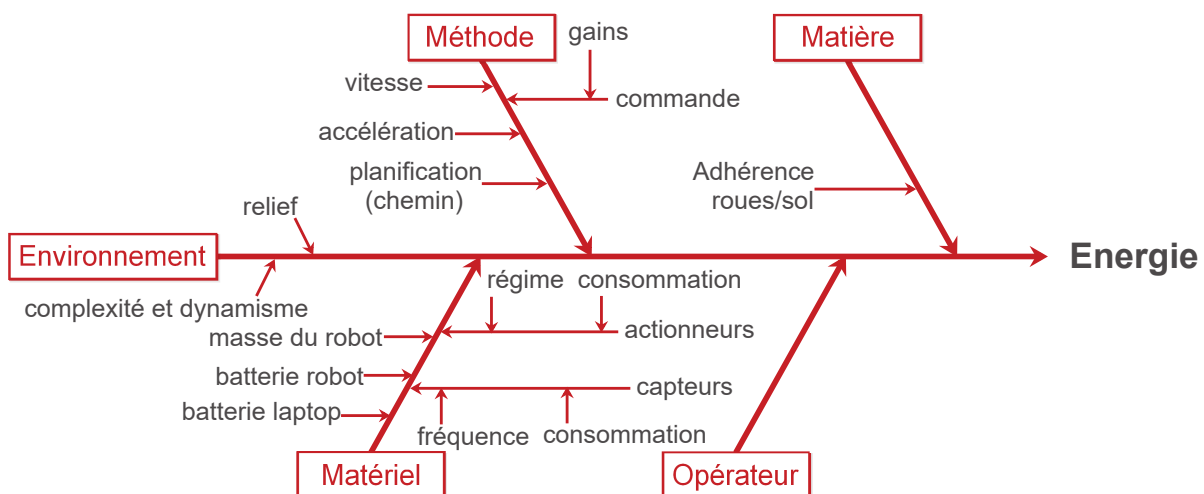


FIGURE 4.8 – Diagramme causes-effet pour l'axe énergie appliqué au système d'étude

Le diagramme d'interdépendance (Figure 4.9) est par la suite utilisé pour illustrer graphiquement les liens d'influence entre les différents nœuds qui le compose. Deux indicateurs de performance énergétique sont à remarquer :

- **Consommation sur la batterie du robot :** Elle est fonction des composants utilisés et alimentés sur la batterie du Pioneer. Ce sont les capteurs dont certains sont à fréquence modifiable et les actionneurs (moteurs à courant continu (MCC)) dont la consommation est fonction de la vitesse de rotation et de la charge à entraîner. L'environnement dans lequel le robot évolue influe aussi sur sa consommation. Son dynamisme et sa complexité peuvent impliquer des manœuvres susceptibles de rallonger la distance à parcourir. De plus, le relief et la nature du sol (carrelage, gravier, etc.) de l'environnement peuvent impliquer une réduction ou une augmentation de la consommation énergétique par rapport à un déplacement sur un plan horizontal d'une nature donnée.

- **Consommation sur la batterie de l'ordinateur embarqué :** La batterie du PC embarqué alimente différents composants internes à ce dernier dont on ne peut pas maîtriser forcément l'activation ou le régime de fonctionnement (e.g. le ventilateur de refroidissement). Cependant l'architecture de contrôle a en fonction des schémas exécutés, un coût CPU. La consommation de ce dernier sera proportionnelle à sa charge. D'autre part, la communication entre les différents composants y compris la plateforme du Pioneer entraîne une consommation énergétique.

La précédente analyse graphique permet de dégager un certain nombre d'inducteurs sur lesquels on peut agir pour gérer l'énergie. Ce sont en premier lieu les capteurs dont l'alimentation peut être désactivée. Cependant leur communication avec l'ordinateur embarqué doit être toujours assurée afin de pouvoir les réactiver au besoin. Les capteurs ont une consommation constante sur la batterie du PC. En second lieu, se sont les MCC qui ont une consommation variable dépendante de la commande, des caractéristiques dynamiques du robot ou de l'environnement extérieur (pente, sol) et du chemin choisi.

D'autre part, la consommation sur la batterie du PC embarquée peut être gérée par le choix des schémas de contrôle puisque la consommation de communication reste constante.

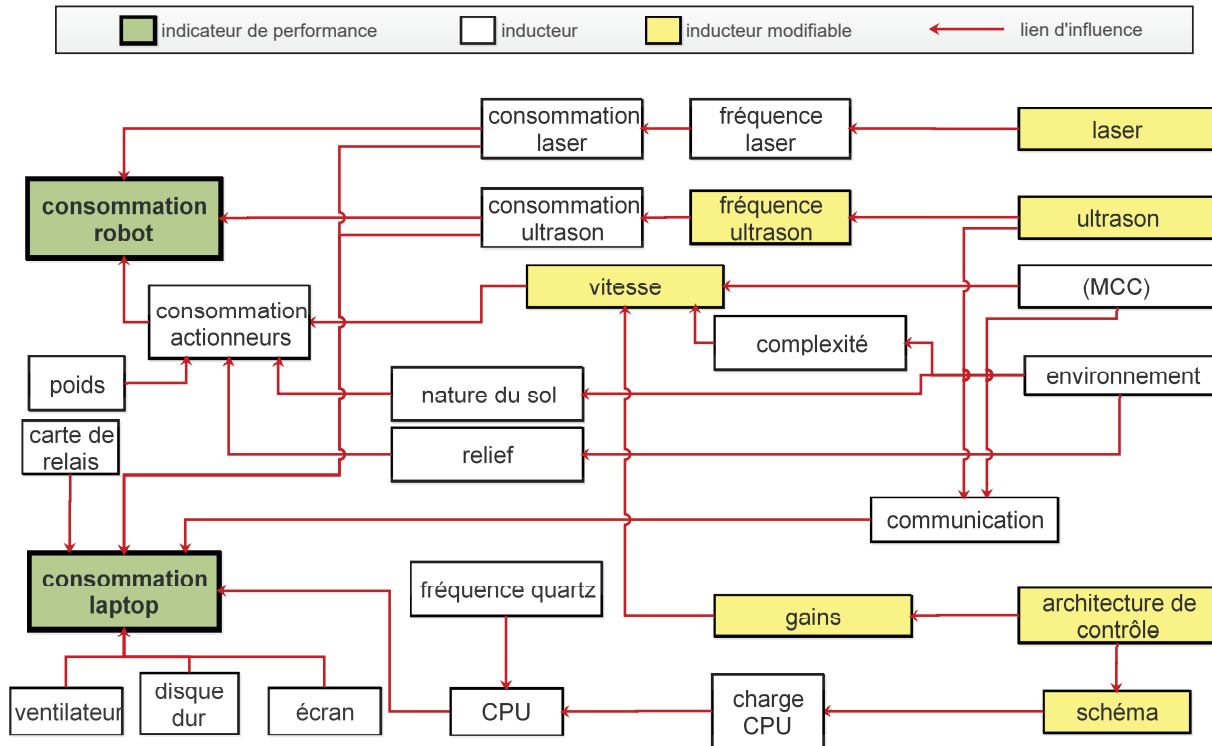


FIGURE 4.9 – Diagramme d'interdépendance de l'axe énergie appliqué au système d'étude

Nous allons établir dans la suite les modèles d'estimation de la consommation énergétique pour les deux indicateurs de performance révélés par l'étude précédente.

4.1.2.2 Modèles d'estimation de l'énergie

Beaucoup de travaux abordent la question d'énergie dans le but de réduire la consommation énergétique, de développer des lois de contrôle en fonction de la consommation ou d'intégrer des contraintes énergétiques parmi les objectifs de la mission. Cependant, de nos jours, peu de travaux étudient en détail l'impact de la configuration matérielle et logicielle sur la consommation énergétique d'un robot et de l'ordinateur embarqué.

La question énergétique étant centrale pour la robotique autonome, on peut considérer qu'elle a été abordée selon trois principaux niveaux :

- **Composant** : L'objectif est de réduire la consommation d'énergie en utilisant des techniques de gestion d'énergie dynamique (*DPM : Dynamic Power Management* [BBD00]). Celles ci sont basées sur l'estimation du niveau d'activité des composants matériels et logiciels afin d'ajuster dynamiquement leur puissance tout en garantissant un certain niveau de performance. L'une des techniques les plus connues de

DPM est le *DVS* (*Dynamic Voltage Scaling*). Elle permet d'adapter dynamiquement la fréquence et le voltage du processeur. Elle est couramment utilisée comme une technique d'économie d'énergie [Lee+12b].

Ces approches ont été étendues et adaptées à la robotique. Par exemples Neto *et al.* ainsi que Mei *et al.* proposent dans [Net+11 ; Mei+05a ; Mei+05b] de désactiver des composants lorsque l'environnement reste inchangé ou que le composant n'est pas utilisé.

Ces techniques orientées gestion matérielle sont complétées par d'autres approches de gestion logicielles comme les algorithmes de *Real-Time Scheduling* (*RTS*), *Rate Monotonic* (*RM*) et *Earliest Deadline First* (*EDF*) qui permettent un ordonnancement des tâches (logicielles) sous des limites temporelles. Dans [Man+14], Mansor *et al.* intègrent ces approches dans le contexte de la robotique mobile en proposant une approche de gestion d'énergie intelligente *Power Management System* (*PMS*). Cette technique permet de maintenir la capacité du robot à réaliser ses tâches du point de vue énergétique. Cependant la charge de calcul est importante et incompatible avec des applications temps réel. L'approche *Joint Speed control and Power* (*JSP*) adresse le problème d'ordonnancement en le considérant comme un problème d'optimisation d'une part de la fréquence du processeur et de la vitesse du robot [ZLH09a ; ZH07 ; BXL06]. Résoudre ce type de problème reste encore irréalisable en temps réel.

- **Système robotique :** La problématique énergétique a été moins traitée au niveau du système robotique. En effet, celle ci est la plupart du temps négligée. Cela peut être légitime pour la robotique industrielle où elle peut être considérée comme illimitée. Mais en robotique mobile cela reste étonnant, surtout pour des missions longues, sans doute en raison de la complexité du problème. Les travaux antérieurs se concentrent principalement sur l'identification de l'énergie consommée le long d'une trajectoire pour une mission mono-tâche. Mei *et al.* propose dans [Mei+05a ; Mei+05b] une analyse expérimentale de la consommation énergétique d'un Pioneer-3DX impliquant son déplacement, les capteurs, le micro-contrôleurs ainsi que le PC embarqué. Plusieurs travaux comme [KK08b ; TKI11b ; TKI14] s'intéressent à la détermination du profil de vitesse optimal pour minimiser la consommation d'énergie le long d'une trajectoire prédéfinie. D'autre part, comme dans [Mei+04], les autres s'intéressent à déterminer la meilleure trajectoire à utiliser du point de vue énergétique pour couvrir une surface donnée. Sadrpour *et al.* ont abordé dans [SJU12 ; SJU13b ; SJU13a] la problématique de résistance au roulement. Ils intègrent ainsi ce facteur pour estimer l'énergie nécessaire pour une mission, connaissant *a priori* les caractéristiques du sol. Enfin, un modèle identifié expérimentalement sur un Khepara III a été proposé par Parasuraman *et al.* dans [Par+14]. Il détaille l'impact des

différents capteurs et composants utilisés. Cependant, la validation expérimentale est faite sur une mission de courte durée et de courte distance en utilisant une seule configuration.

- **Mission :** Peu de travaux s'intéressent à la problématique de l'énergie au niveau mission. Ils concernent souvent des applications simples et mono-tâches. [Sie+03] considère la problématique d'énergie comme centrale dans le cadre de mission sur la lune réalisée par un système de robots collaboratifs. Sierhuis *et al.* identifient l'énergie à chaque sous-système pour réaliser la mission. Cependant, les modèles utilisés ne sont pas détaillés. Une approche de gestion de l'énergie par l'adaptation des tâches et des environnements est proposée dans [Oga+06] mais son implémentation n'est pas explicitée. [ZLV07] et [OZ00] ont traité le problème de recharge énergétique dans des missions robotiques que ce soit par un dispositif ou par un robot dédié pour cette tâche. Lesage et Langoria abordent cet axe de performance dans [LL15] du point de vue probabiliste en estimant la faisabilité d'une mission robotique du point de vue énergétique.

Il faut maintenant détailler l'architecture matérielle du système robotique étudié. La figure 4.10 montre les éléments alimentés par la batterie du robot (noir) et par la batterie de l'ordinateur embarqué (blanc). Les éléments qu'on peut activer/désactiver sont en pointillé alors que les éléments qui ont une consommation répartie sur les deux sources énergétiques sont en gris.

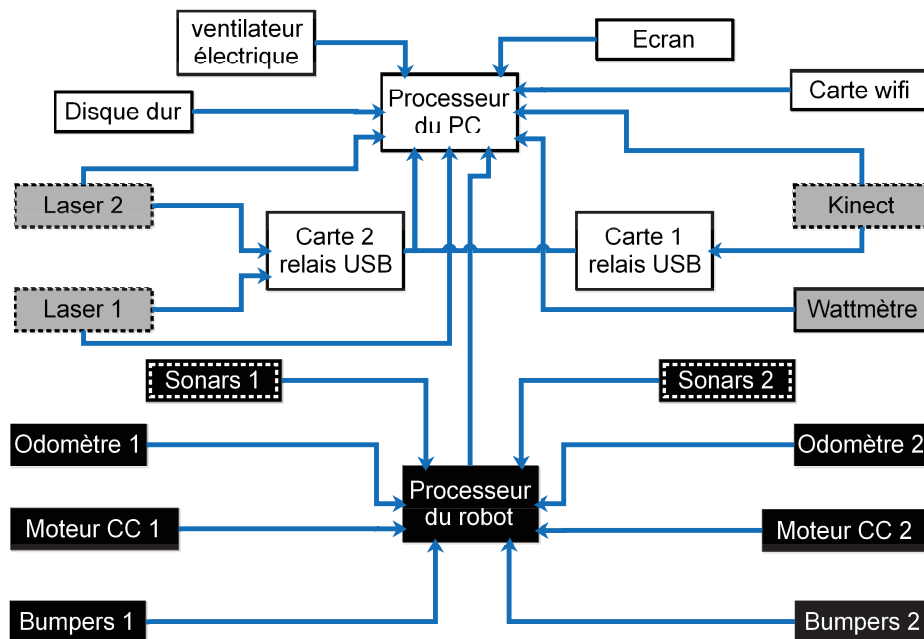


FIGURE 4.10 – Architecture matérielle du système robotique étudié

On note les indicateurs de performance énergétique $Perf_{SR, Energie}^{Robot}$ pour l'énergie consommée sur la batterie du robot et $Perf_{SR, Energie}^{PC}$ pour celle consommée sur la batterie du PC. Nous allons expliciter le modèle d'estimation énergétique de ces deux batteries dans ce qui suit.

a. Estimation de la consommation énergétique sur la batterie du robot

a.1. Travaux existants

Trois principales méthodes, ont été proposées pour déterminer les modèles de consommation d'énergie de robot différentiels non-holonomes comme le Pioneer 3DX :

- **A1** : Approche basée sur la modélisation électrique et mécanique du véhicule,
- **A2** : Approche basée sur la modélisation dynamique du véhicule,
- **A3** : Construction de modèles de lois empiriques à partir de mesures expérimentales.

Bien que **A1** permet d'établir une modélisation détaillée de la consommation énergétique d'un système mobile, elle n'offre pas de résultats meilleurs que ceux établis par **A2** et **A3**. Nous détaillerons dans la suite les principaux travaux adoptant ces deux approches sur lesquelles se basera notre modèle proposé.

Sadrpour *et al.* [SJU12 ; SJU13b ; SJU13a] ont étudié la consommation énergétique d'un véhicule télé-opéré. Le modèle proposé (4.9) est basé sur l'approche établie dans [UPÇ12].

$$P(t) = (W \cdot \sin(\theta(t)) + f \cdot W \cdot \cos(\theta(t))) + m \cdot a(t) + C_I \cdot v(t) + b + \varepsilon(t) \quad (4.9)$$

Où $P(t)$ est la puissance à l'instant t , $v(t)$ est la vitesse, $a(t)$ l'accélération, $\theta(t)$ est la pente de la route, f est le coefficient de résistance au roulement, m est la masse du robot, W est son poids, C_I la résistance dûe aux frottements internes, b correspond à toute autre source constante de consommation énergétique comme les capteurs et $\varepsilon(t)$ l'erreur de modélisation.

(4.9) peut être linéarisée pour des pentes ($\theta(t) < 15^\circ$) pour donner l'équation suivante :

$$P(t) = W \cdot \theta(t) + f \cdot W + m \cdot a(t) + C_I \cdot v(t) + b + \varepsilon(t) \quad (4.10)$$

Ce modèle montre l'impact du sol sur lequel le robot se déplace sur la consommation d'énergie. Il exprime également qu'en plus de l'énergie de déplacement, d'autres sources additionnelles de consommation énergétique peuvent être considérées.

Mei *et al.* réalisent dans [Mei+05b] une analyse expérimentale détaillée des différentes sources de consommation énergétique d'un Pioneer-3DX. Ils proposent ainsi un modèle d'estimation de la consommation nécessaire au déplacement (4.11) issu de (4.10) en négligeant l'impact du sol sur la consommation énergétique.

$$P(t) = (f \cdot W + m \cdot a(t))v(t) + P_l \quad (4.11)$$

Avec P_l la puissance dissipée par frottement dans les moteurs et par résistance aérodynamique du robot. Ces pertes sont supposées constantes tandis qu'elles dépendent de la vitesse comme le montre (4.10). L'étude expérimentale a vérifié que pour une le modèle linéaire (4.12) obtenu à partir de (4.11) était acceptable.

$$P(v) = A \cdot v + B \quad (4.12)$$

Avec A et B les constantes de la droite identifiée expérimentalement.

D'autres travaux modélisent la consommation électrique à partir des lois électriques et mécaniques. Kim et Kim [KK08b] proposent une analyse complète de la consommation énergétique d'un Pioneer-3DX afin de déterminer le profil de vitesse minimisant l'énergie. La consommation d'énergie de deux MCC identiques pour un mouvement de ligne droite est uniquement considérée. L'effet aérodynamique est négligé. De plus, une analyse électromécanique détaillée des pertes énergétiques des moteurs à courant continu est proposée dans [SSP02]. Dans la même logique, Tokekar *et al.* développent dans [TKI14 ; TKI11b] une méthode permettant d'optimiser la vitesse de façon à minimiser l'énergie de locomotion pour des trajectoires plus complexes. Un modèle de consommation énergétique est ainsi proposé (4.13) [Mot77].

$$E = \int_0^{t_f} [C_1 \cdot a(t)^2 + C_2 \cdot v(t)^2 + C_3 \cdot v(t) + C_4 + C_5 \cdot a(t) + C_6 \cdot a(t) \cdot v] dt \quad (4.13)$$

C_1 à C_6 représentent des combinaisons des paramètres moteurs. Cependant (4.13) peut être fortement simplifiée en (4.14) pour des vitesses constantes $v = \text{cte}$ (accélération nulle).

$$E = \int_0^{t_f} [C_2 \cdot v(t)^2 + C_3 \cdot v(t) + C_4] dt \quad (4.14)$$

Les travaux qui s'intéressent à la formulation la consommation énergétique d'un Pioneer-3DX se focalisent généralement sur la locomotion et donc la consommation des MCC. Cependant, les autres composants présents sur le robot (capteurs) présentent une source de consommation non négligeable et qui doit être explicitement prise en compte. Sadrpour *et al.* identifient cette source de consommation externe (au moteur) par une constante sans en détailler le calcul. Mei *et al.* abordent le problème des sources de consommation ne relevant pas la locomotion. Cependant cet impact n'est pas évalué pour les différentes configurations matérielles et logicielles pouvant être utilisées lors d'une mission de longue durée. Généralement, ces travaux concernent des missions de courte durée et pour de faible distance parcourue..

a.2. Modèle d'estimation proposé

Nous adaptons dans la suite la formulation proposée par Parasuraman *et al.* dans [Par+14] pour le robot Khepera III dans le but d'obtenir un modèle global de la consommation énergétique considérant les différentes configurations matérielles et logicielles du robot [Jai+16a]. Ils caractérisent la puissance consommée par les différents composants du robot comme les modules de Wifi, Bluetooth et pour différentes charges utiles. Le modèle de consommation de locomotion proposé provient de (4.14). Les résultats ont été validés sur une mission mono-tâche et de courte distance ($< 7\text{m}$).

Pour une Alternative d'Implémentation AI donnée, la consommation électrique instantanée sur la batterie du robot correspond à $P_{BR}(AI)$. Cette puissance peut être décomposée en une partie dynamique et une partie statique. La partie dynamique est la somme des puissances $P_{BR_{D_{yn} \ i}}$ des i composants dont la puissance consommée peut évoluer dans le temps en fonction des paramètres de configuration. Citons l'exemple de la consommation électrique des moteurs qui dépend de la vitesse ainsi que celles des sonars où la fréquence d'utilisation est réglable. La partie statique $P_{BR_{Stat \ j}}$ dénote la consommation électrique des j composants consommant une puissance constante lorsqu'ils sont recrutés. C'est le cas de certains capteurs (laser, kinect) ou des dispositifs de communication. Selon l'AI et les composants correspondants recrutés, la consommation instantanée $P_{BR}(AI)$ peut significativement changer, comme le dénote (4.15).

$$P_{BR}(AI) = \sum_{i=1}^{n_1} \alpha_i \cdot P_{BR_{D_{yn} \ i}} + \sum_{j=1}^{n_2} \beta_j \cdot P_{BR_{Stat \ j}} \quad (4.15)$$

Avec n_1 est le nombre de composants dynamiques, n_2 le nombre de composants statiques, α_i et β_j sont égaux à 1 si le composant i/j est utilisé dans l'AI considéré et 0 autrement.

La consommation énergétique $E_{BR}(AI)$ d'une AI (4.16) est classiquement obtenue en multipliant la consommation électrique instantanée par la durée d'activité T de l'AI correspondante.

$$E(AI) = P(AI) \cdot \Delta T(AI) . \quad (4.16)$$

Nous détaillons maintenant les puissances consommées par les différents composants de notre robot.

-Composants statiques

Le tableau 4.2 montre la puissance consommée par les différents composants statiques alimentés par la batterie du robot. Les différentes puissances ont été mesurées par le wattmètre embarqué en activant les composants correspondants séparément.

Tableau 4.2 – Puissances consommées par les composants statiques

Composant statique	Puissance (W)
Camera Kinect	$P_{BR_{Kinect}} = 2.82$
Contrôleur	$P_{BR_{Controller1}} = 2.67$
Laser	$P_{BR_{Laser}} = 2.34$

–Composants dynamiques : Les moteurs à courants continus

Les MCC sont des composants dont la consommation électrique dépend, comme on l'a vu précédemment, de plusieurs paramètres. Nous allons reprendre dans ce qui suit l'équation (4.13) dont le modèle de consommation des moteurs peut être définie par :

$$P_{BR_{locomotion}}(a, v) = C_1 \cdot a^2 + C_2 \cdot v^2 + C_3 \cdot v + C_4 + C_5 \cdot a + C_6 \cdot a \cdot v . \quad (4.17)$$

Souvent la trajectoire du robot tout au long d'une mission peut être décomposée en une suite de lignes droites où le robot peut rouler à vitesse constante. Cette hypothèse a été souvent adoptée dans la littérature [SJU13b; SJU13a]. La prise en compte de la dynamique du système, abordés dans [UPÇ12] donne des modèles complexes, difficiles à manipuler. En utilisant cette hypothèse, le modèle de puissance de mouvement (4.17) peut être simplifié en (4.18). Ainsi à partir de cette dernière, il nous reste à identifier C_2 , C_3 et C_4 .

$$P_{BR_{locomotion}}(v) = C_2 \cdot v^2 + C_3 \cdot v + C_4 \quad (4.18)$$

La figure 4.11 montre les résultats expérimentaux de la puissance consommée pour un déplacement linéaire du robot à différentes vitesses. En projetant (4.18) sur le relevé expérimental, la loi de consommation des MCC est déterminée et peut être exprimée par (4.19) avec un coefficient de détermination égal à 0.99.

$$P_{BR_{locomotion}}(v) = 6.25 \cdot v^2 + 9.79 \cdot v + 3.66 \quad (4.19)$$

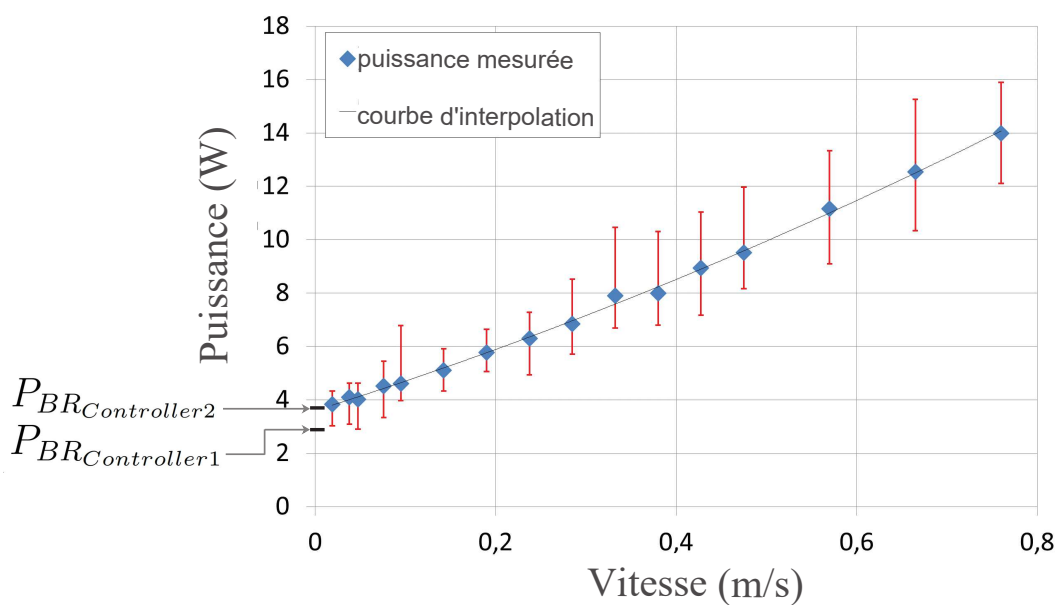


FIGURE 4.11 – Puissance de locomotion pour différentes vitesses du robot

On peut remarquer que lorsque le robot est à l'arrêt ($v = 0$ m/s), la puissance est égale à C_4 (3,66 W). Cette puissance $P_{BR_{Controller2}}$ correspond à la consommation requise par les différentes cartes électroniques de bas niveau y compris le micro-contrôleur au sein du robot.

De plus, il faut remarquer que, normalement $P_{BR_{Controller2}}$ doit être égal à $P_{BR_{Controller1}}$. $P_{BR_{Controller2}}$ est en fait obtenue par extrapolation. Alors que $P_{BR_{Controller1}}$ est mesurée même si on applique une commande de mouvement en utilisant $v = 0$ m/s. Imposer C_4 à $P_{BR_{Controller1}}$ de l'équation (4.18) va réduire le coefficient de détermination à 0.97 et diminuer ainsi la précision de l'estimation. Nous proposons alors d'utiliser la formulation suivante :

$$\left\{ \begin{array}{ll} P_{BR_{locomotion}}(v) &= P_{BR_{Controller1}} \\ &= 2.67 \quad \text{si } v = 0 \\ P_{BR_{locomotion}}(v) &= 6.25 \cdot v^2 + 9.79 \cdot v + P_{BR_{Controller2}} \\ &= 6.25 \cdot v^2 + 9.79 \cdot v + 3.66 \quad \text{si } v \neq 0 \end{array} \right. \quad (4.20)$$

Si l'on s'intéresse dans la suite au déplacement du robot, il est possible de déduire à partir des équations (4.16) et (4.19) l'énergie estimée pour parcourir une distance d à une vitesse v par l'équation suivante.

$$E_{BR_{locomotion}}(d, v) = 6.25 \cdot d \cdot v + 9.79 \cdot d + 3.66 \cdot \frac{d}{v}. \quad (4.21)$$

La figure 4.12 représente l'énergie de locomotion nécessaire en fonction de la vitesse pour parcourir trois distances (50, 100 et 300 m). Elle montrent que la consommation d'énergie augmente considérablement à faible vitesse. De plus, les courbes d'énergie passent par un minimum. Il est donc possible de déterminer une vitesse optimale v_{opt} (4.22).

$$\frac{dE_{BR_{locomotion}}(d, v)}{dv} \Rightarrow v_{opt} = \sqrt{\frac{3.66}{6.25}} = 0.76 \text{ m/s} \quad (4.22)$$

Trois zones peuvent être distinguées dans la figure 4.12 :

- Z1 où $v < v_{opt}$ où l'énergie nécessaire pour la locomotion est supérieure à l'énergie minimale $E_{min}(v_{opt})$.
- Z2 où $v_{opt} < v \leq v_{r_{max}}^{exp}$ et l'énergie de locomotion est supérieure à $E_{min}(v_{opt})$. Pour notre robot, la vitesse maximale $v_{r_{max}}^{th}$ annoncée par le fabricant ne peut pas être pratiquement atteinte. Nous constatons que la vitesse maximale atteignable par le robot tombe expérimentalement de $v_{r_{max}}^{th} = 1,2 \text{ m/s}$ à $v_{r_{max}}^{exp} = 0,76 \text{ m/s}$ à cause du vieillissement de la plateforme.
- Z3 où $v_{r_{max}}^{exp} < v \leq v_{r_{max}}^{th}$. Cette zone peut être étudiée en théorie, mais ne peut être utilisée expérimentalement.

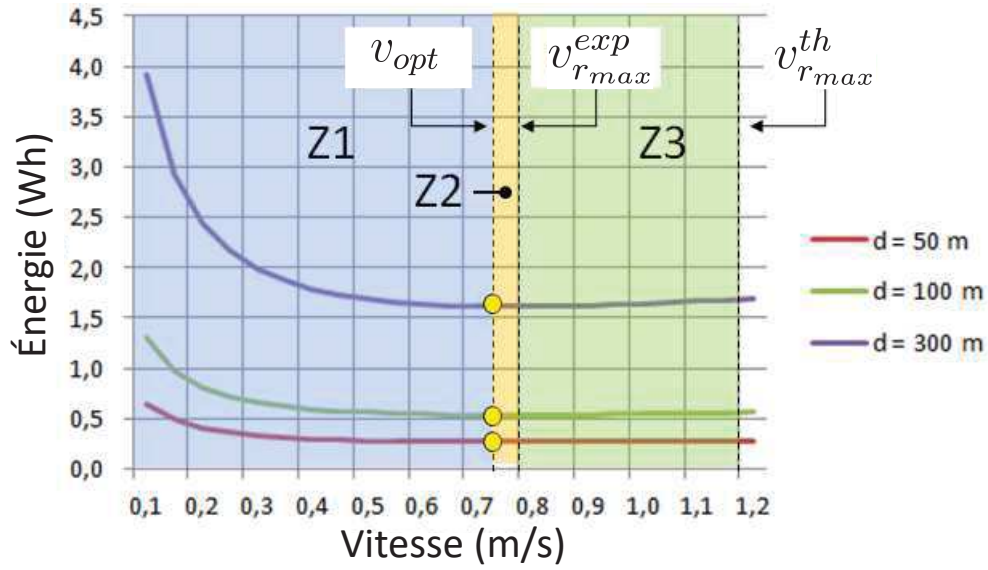


FIGURE 4.12 – Énergie de locomotion pour différentes vitesses et distances parcourues

Revenons maintenant à la contrainte de discrétisation imposée pour la prise de décision (voir section 2.2.3.2). La figure 4.13 montre une illustration du principe de discrétisation pour la courbe de consommation énergétique établie. Le choix des valeurs pertinentes pour une allure de consommation énergétique convexe tel notre cas et présentant un minimum à v_{opt} se fait en fonction des vitesses maximale $v_{A_{i_{max}}}$ et minimale $v_{A_{i_{min}}}$ possibles pour une activité donnée. Trois cas sont alors possibles :

- **cas 1** : si $[v_{A_{i_{min}}}, v_{A_{i_{max}}}] < v_{opt}$ alors l'ensemble de vitesses pertinentes se réduit au singleton $\{v_{A_{i_{max}}}\}$. Cette vitesse correspond à la fois au minimum d'énergie et minimise la durée d'exécution de l'activité. C'est pratiquement le cas du robot utilisé dont la vitesse optimale est très proche de sa vitesse maximale.
- **cas 2** : si $v_{opt} \in [v_{A_{i_{min}}}, v_{A_{i_{max}}}]$ alors l'ensemble de vitesses pertinentes correspond au doublet $\{v_{opt}, v_{A_{i_{max}}}\}$. v_{opt} est évidemment intéressante puisqu'il minimise l'énergie pour une AI donnée. D'autre part, $v_{A_{i_{max}}}$ minimise la durée.
- **cas 3** : si $v_{opt} < [v_{A_{i_{min}}}, v_{A_{i_{max}}}]$ alors l'ensemble de vitesses pertinentes correspond au triplet $\{v_{A_{i_{min}}}, v_{cmp}, v_{A_{i_{max}}}\}$. v_{cmp} présente le meilleur compromis entre la consommation énergétique et la durée. Et puisque ce cas correspond à la zone croissante de la courbe, $v_{A_{i_{min}}}$ correspond à l'énergie minimale possible et $v_{A_{i_{max}}}$ minimise la durée de l'activité.

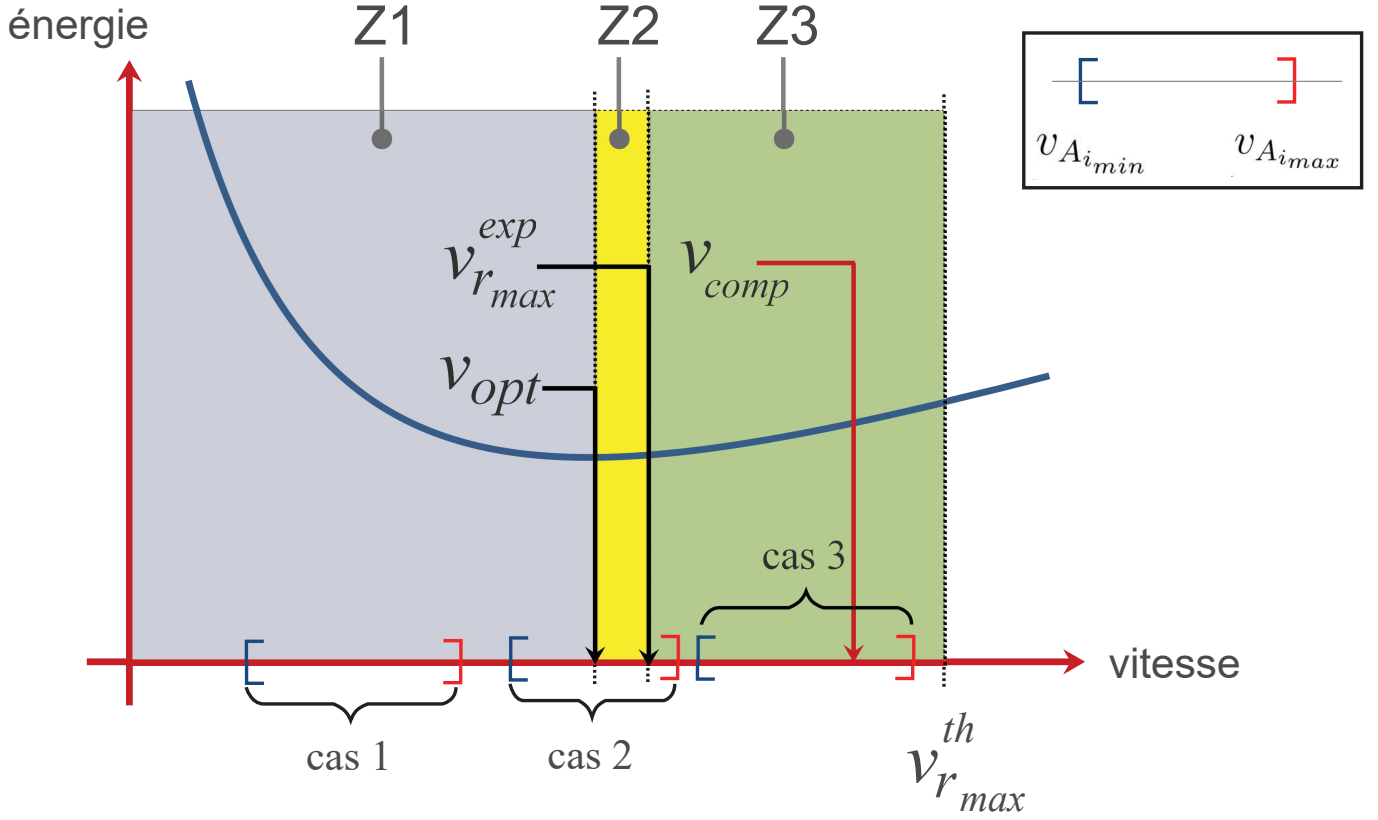


FIGURE 4.13 – Principe de discrétisation et de choix de la vitesse

Le modèle d'estimation de l'énergie consommée pour le déplacement (MCC) est maintenant établi, passons à un autre type composant dynamique : les sonars.

-Composants dynamiques : Les sonars

Les sonars sont aussi des composants dont la consommation énergétique dépend d'un paramètre (fréquence d'utilisation). La figure 4.14 montre les relevés expérimentaux de puissance pour différentes fréquences f .

Les points expérimentaux de la figure 4.14 peuvent être approchés par un polynôme quadratique de coefficient de détermination égal à 0,99. Dans nos expérimentations, nous conserverons une fréquence constante à 25 Hz, ce qui correspond à $P_{BR_{US}}(25) = 0,037$ W.

$$P_{BR_{US}}(f_{US}) = 4 \cdot 10^{-5} \cdot f_{US}^2 + 5.1 \cdot 10^{-4} \cdot f_{US} \quad (4.23)$$

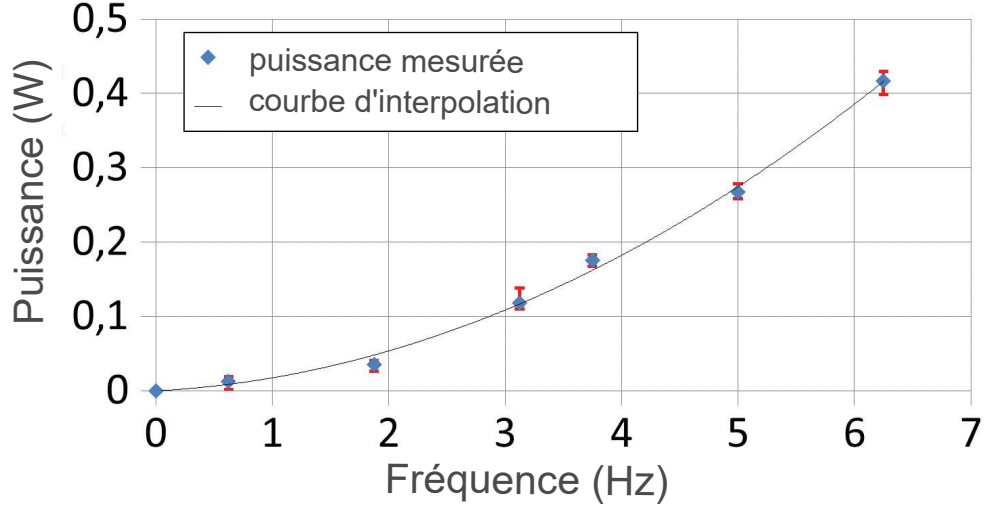


FIGURE 4.14 – Consommation énergétique des sonars pour différentes fréquences

Dans [Mei+05b], un modèle d'estimation de la consommation des sonars a été aussi proposé. Cependant, il suit une loi linéaire et non pas quadratique. La courbe expérimentale donnée ne passe pas par l'origine pour une fréquence nulle ce qui peut paraître étrange. Toutefois, les conditions expérimentales ne sont pas expliquées en détail dans cet article.

-Formulation de la consommation énergétique d'une AI sur la batterie du robot

A partir de la formulation proposée dans (4.15) et de l'estimation de la consommation énergétique des composants dynamiques et statiques, il est possible d'établir le modèle d'estimation global/unifié suivant, pour l'ensemble du système robotique cible :

$$P_{BR}(AI) = \alpha_1 P_{BR_{locomotion}}(v) + \alpha_2 \cdot P_{BR_{US}}(f_{US}) + \beta_1 \cdot P_{BR_{Kinect}} + k_1 \cdot \beta_2 \cdot P_{BR_{Laser}} \quad (4.24)$$

Où $k_2 \in \{0, 1, 2\}$ est le nombre de lasers utilisés, α_i (dynamiques) et β_i (statiques) sont des coefficients booléens qui indiquent si le composant i correspondant est utilisé ou non.

Si on suit le même raisonnement que pour (4.22), à partir de la formulation générique de (4.24) et de (4.16), on peut déduire une formulation générique de la vitesse optimale pour une AI (4.25).

$$V_{opt}(AI) = \sqrt{\frac{3.66 + \alpha_2 \cdot P_{BR_{US}}(f_{US}) + \beta_1 \cdot P_{R_Kinect} + k_1 \cdot \beta_2 \cdot P_{BR_{Laser}}}{6.25}} \quad (4.25)$$

On peut remarquer que l'ajout de capteurs (voir 4.22) ne peut qu'augmenter la vitesse optimale minimale. Cependant ces vitesses (pour des AI mobilisant des capteurs) ne peuvent être atteintes car le robot est limité pratiquement à $v_{r_{max}}^{exp}$ 0.76 m/s.

La totalité des puissances pour les différentes AI possibles peut être trouvée dans l'Annexe A avec une mise en avant des extrêmes par AI. Le tableau 4.3 montre, à titre d'exemple, les puissances instantanées consommées par le système robotique pour des AI permettant une avance simple (AS) du robot avec différentes combinaisons de capteurs. La vitesse est fixée à 0.5 m/s. Selon l'AI considérée, la puissance peut aller de 10.12 W jusqu'à 17.66 W pour les 12 exemples d'AI avec une différence maximale de 74.5 %. Il est clair que l'influence des consommations des capteurs et actionneurs choisis sur la puissance instantanée est loin d'être négligeable.

Tableau 4.3 – Puissance instantanée estimée pour différentes AI à $v_r = 0.5$ m/s

	US	Laser	Kinect	MCC (0.5 m/s)	$P_{BR}(AI)$ (W)
AS	1	2	1	1	17.66
	0	2	1	1	17.62
	1	1	1	1	15.32
	0	1	1	1	15.28
	1	2	0	1	14.84
	0	2	0	1	14.80
	1	0	1	1	12.98
	0	0	1	1	12.94
	1	1	0	1	12.50
	0	1	0	1	12.46
	1	0	0	1	10.16
	0	0	0	1	10.12

Il faut aussi remarquer que la consommation sur la batterie du robot peut être la même pour différentes AI dont les ressources matérielles sont les mêmes, mais les algorithmes les exploitant étant différents, l'impact sur la batterie du robot sera différent.

Si toutes les configurations de systèmes de contrôle proposées sont considérées, la consommation d'énergie peut varier de 4,7 W (rotation sur place sans capteurs) jusqu'à 17,66 W lorsque tous les capteurs sont mobilisés (voir Annexe A). Cela montre l'import-

tance de la prise en compte des différentes configurations matérielles et logicielles du robot et que cette dernière ne peut être négligée ou considérée comme constante.

b. Estimation de la consommation énergétique sur la batterie du PC

La question de l'énergie consommée sur la batterie d'ordinateur portable supportant l'architecture de contrôle ainsi que les différentes communications avec les capteurs robot est très rarement adressée.

L'énergie fournie par la batterie de l'ordinateur portable dépend de trois principaux facteurs :

- Le processeur de l'ordinateur portable.
- Les composants internes comme le disque dur et les cartes de communication (wifi, Bluetooth, etc).
- Les composants externes tels que l'écran d'affichage et les connexions avec les périphériques via le lien USB.

La puissance consommée par le processeur $P_{PC_{Proc}}$ a été largement étudiée pour des puces CMOS. Elle peut être décomposée en parties dynamiques et statiques (4.26). La consommation dynamique $P_{proc_{dyn}}$ est celle dissipée au cours des activités de transition du processeur, cette énergie peut être modélisée par une loi cubique en fonction de la fréquence d'horloge du CPU [Bro+00]. La puissance statique (ou puissance de veille) $P_{proc_{stat}}$ correspond à la consommation d'énergie lorsque le processeur n'a pas de tâche à exécuter. Cette puissance est nettement inférieure à la puissance dynamique, mais ne peut pas être considérée comme nulle. En supposant que la puissance de veille est constante, Xu *et al.* dans [Xu+04] propose la formulation suivante :

$$P_{PC_{Proc}} = P_{proc_{dyn}} + P_{proc_{stat}} = A \cdot f^3 + B \quad (4.26)$$

Où A et B sont deux constantes.

Ce modèle d'estimation de la consommation du processeur est utilisé dans [ZH07; ZLH09b] pour contrôler la fréquence du processeur lors d'une tâche de reconnaissance d'un Pioneer-3DX. Le même modèle a été utilisé dans [BXL06] pour contrôler la fréquence du processeur afin de réduire la consommation d'énergie lors d'un déplacement avec évitement d'obstacles.

Ce modèle initial de la consommation du processeur peut être affiné [Kim+14] en considérant que le courant de veille est constant et que la tension d'alimentation et la

fréquence d'horloge sont proportionnelles en remplaçant B dans (4.26) par $P_{proc_{stat}} = C \cdot f$. D'autres puissances supplémentaires peuvent être considérées telles que la consommation des mémoires vives ou de la mémoire cache $P_{proc_{cache}}$ (4.27).

$$P_{PC_{Proc}} = P_{proc_{dyn}} + P_{proc_{stat}} + P_{proc_{cache}} = A \cdot f^3 + C \cdot f + P_{proc_{cache}} \quad (4.27)$$

Les composants internes du PC embarqué impactent aussi la consommation énergétique. Cependant, ces composants tels que le disque dur ou les cartes de communication (1 W dans [Soh+10 ; SJU12]) ont une consommation pratiquement constante [BJ12].

D'autre part, les connexions des périphériques externes impactent aussi la consommation énergétique. Cela concerne principalement les connexions USB avec les périphériques externes tels que les capteurs (laser, kinect), les cartes relais ou le contrôleur du robot. En fonction du type du périphérique le courant peut aller jusqu'à 0.5 A [KB02] sur les ports USB 1.x et 2.0 et jusqu'à 0.9 A sur les ports USB 3.x. Cela correspond à des puissances allant de 2.5 W jusqu'à 4.5 W.

A partir de cette analyse, il est clair que la batterie du PC embarqué alimente plusieurs composants internes et externes. Cependant, leur consommation dépend fortement de la configuration matérielle et logicielle de l'AI mise en œuvre ainsi que du code exécuté.

Nous détaillons dans la suite l'analyse des différentes consommations (internes et externes) observées à partir de mesures expérimentales réalisées sur des durées de 60 s.

b.1. Composants internes

L'ordinateur portable intègre de nombreux périphériques internes consommant de l'énergie. Cette consommation interne dépend de nombreux facteurs tels que le code exécuté, l'accès au disque dur, le ventilateur, la consommation de la carte wifi et bien sûr le CPU. Malheureusement, il est difficile de dissocier les effets de tous ces éléments. Cependant, comme supposé dans [Oga+06 ; Mei+05a ; Mei+05b], tous ces facteurs peuvent être intégrés dans la consommation d'énergie du processeur. Ils ont considéré, qu'en première approximation, cette puissance est constante pour une AI donnée.

Le tableau 4.4 montre la consommation énergétique du processeur pour différentes configurations matérielles d'un schéma d'Avance Simple. La puissance varie pour ces exemples de 10,73 W à 13,34 W. Ainsi, en fonction des capteurs et algorithmes utilisés, la consommation du processeur des AI d'un schéma d'Avance Simple peut varier jusqu'à 24,3 %.

Tableau 4.4 – Puissance instantanée du processeur pour différentes configurations de AI avec Avance Simple

	US	Laser	Kinect	MCC (0.5 m/s)	$P_{PC_{proc}}(AI)$ (W)
AS	1	2	1	1	13.33
	0	2	1	1	13.27
	1	1	1	1	13.34
	0	1	1	1	13.28
	1	2	0	1	11.54
	0	2	0	1	11.48
	1	0	1	1	12.78
	0	0	1	1	12.72
	1	1	0	1	10.87
	0	1	0	1	10.73
	1	0	0	1	11.43
	0	0	0	1	10.97

Nous considérons l'écran du PC comme un périphérique interne de l'ordinateur portable puisqu'il n'a pas besoin d'une connexion USB supplémentaire. La consommation d'énergie de l'écran peut être facilement mesurée en calculant la différence entre la consommation lorsqu'il est en marche puis éteint. La $P_{PC_{ecran}}$ mesurée est égale à 2,69 W. La consommation de l'écran utilise de 13,1% à 18,7% de la consommation d'énergie totale de l'ordinateur portable. Il est évident qu'elle représente une importante partie de la consommation d'énergie. L'écran de l'ordinateur sera donc éteint au cours d'une mission afin de diminuer la consommation énergétique prélevée sur la batterie du PC.

b.2. Composants externes

Même si les différents capteurs sont alimentés par la batterie du robot, la connexion USB avec le PC embarqué implique une consommation sur sa batterie. Les cartes des relais sont quant à elles complètement alimentées par le PC (Tableau 4.5).

b.3. Consommation globale sur la batterie de l'ordinateur embarqué

La consommation totale est maintenant considérée. Le tableau 4.6 montre la puissance totale délivrée par la batterie du PC pour les AI d'une Avance Simple. Les puissances (avec affichage de l'écran) varient de 11.71 W jusqu'à 17.02 W avec un écart de 45 %.

Tableau 4.5 – Consommation des périphériques externes de l'ordinateur embarqué

Périphériques externes	Noté	Puissance (W)		
Laser	$P_{PC_{laser}}$	0.447		
Controleur	$P_{PC_{controleur}}$	0.075		
Kinect	$P_{PC_{kinect}}$	1.180		
Wattmetre	$P_{PC_{Watt}}$	0.490		
Carte relais	$P_{PC_{CR1}}$	aucun relais	1 relais	2 relais
	$P_{PC_{CR2}}$	0.087	0.405	0.720

Tableau 4.6 – Puissance instantanée globale consommée sur la batterie du PC pour différentes configurations de AI avec une Avance Simple

	US	Laser	Kinect	MCC (0.5 m/s)	$P_{PC}(AI)$ (W)
AS	1	2	1	1	17.02
	0	2	1	1	16.96
	1	1	1	1	16.31
	0	1	1	1	16.25
	1	2	0	1	13.71
	0	2	0	1	13.65
	1	0	1	1	15.04
	0	0	1	1	14.98
	1	1	0	1	12.33
	0	1	0	1	12.19
	1	0	0	1	12.17
	0	0	0	1	11.71

Si nous considérons maintenant les différentes AI utilisées, la consommation d'énergie sur la batterie de l'ordinateur portable varie de 11,71 W (Avance Simple sans capteurs) jusqu'à 17,84 W (centrage US, 2 lasers et la kinect) (voir Annexe A). Cet écart considérable montre que la consommation sur la batterie du PC ne peut être considérée constante quelle que soit l'AI mise en œuvre.

b.4. Formulation de la consommation énergétique d'une AI sur la batterie du PC

De même que pour la batterie du robot (4.15), la puissance consommée sur la batterie du PC $P_{PC}(AI)$ peut être divisée en une partie dynamique et une partie statique . (4.28).

$$P_{PC}(AI) = \sum_{i=1}^{n_3} \alpha_i \cdot P_{PC_{Dyn_i}} + \sum_{j=1}^{n_4} \beta_j \cdot P_{PC_{Stat_j}} \quad (4.28)$$

Avec n_3 le nombre composants dynamiques et n_4 le nombre de composants statiques, α_i et β_j sont des coefficients booléens indiquant si le composants i/j est utilisé ou non dans l'AI correspondante.

Pour une AI donnée, on peut établir une formulation du modèle d'estimation de l'énergie consommée sur la batterie du PC en fonction de la configuration des périphériques externes (4.29).

$$P_{PC}(AI) = P_{PC_{Proc}}(AI) + P_{PC_{contrôleur}} + P_{PC_{Watt}} + \beta_1 \cdot P_{PC_{ecran}} + \beta_2 \cdot P_{PC_{kinect}} + k_1 \cdot \beta_3 \cdot P_{PC_{laser}} + \beta_4 \cdot P_{PC_{CR_1}}(k_2) + \beta_5 \cdot P_{PC_{CR_2}}(k_3)$$

(4.29)

Avec :

β_1 à β_5 sont des coefficients booléens qui sont à 1 si la connexion est établie avec le PC et 0 si non.

$k_1 \in \{1, 2\}$ représente le nombre de lasers connectés au PC.

$k_2 \in \{0, 1, 2\}$ représente le nombre de lasers connectés (carte relais 1).

$k_3 \in \{0, 1\}$ représente si la kinect est connectée ou non (carte relais 2). On peut avoir un troisième état (2) si l'on ajoute ultérieurement un autre périphérique sur cette carte.

c. Validation expérimentale des modèles d'estimation d'estimation de l'énergie

Pour valider expérimentalement les modèles d'estimation de la consommation énergétique sur les batteries du robot et du PC, une mission simple de déplacement sur une ligne droite de 20 m et 50 m a été réalisée. Comme d'autres travaux le confirment, une mission robotique peut souvent se décomposer en une séquence de lignes droites [BCP95b ; Mei+04 ; KK08b].

La précision de l'estimation est évaluée en calculant l'erreur de prédiction (4.30) entre l'énergie mesurée E_{mes} et celle prédite E_{pred} .

$$Erreur(\%) = \frac{E_{mes} - E_{pred}}{E_{mes}} \cdot 100 \quad (4.30)$$

L'expérimentation proposée consiste à mettre en œuvre le schéma d'Avance Simple

(AS) pour différentes combinaisons de capteurs alimentés (non utilisés pour l'évitement d'obstacle). La même vitesse est appliquée sur les deux roues en supposant que le robot se déplace en ligne droite dans un espace ouvert sans aucun obstacle.

c.1. Batterie robot

La figure 4.15(b) montre l'énergie consommée avec des vitesses allant de 0.1 à 0.76 m/s pour des AI sans capteurs, avec deux lasers et la kinect. Les courbes d'estimation correspondantes (trait pour l'AI sans capteurs et tiret pour les AI avec des capteurs). Les erreurs de prédiction calculées avec (4.30) sont montrées dans la figure 4.15(a) ainsi que le tableau 4.7.

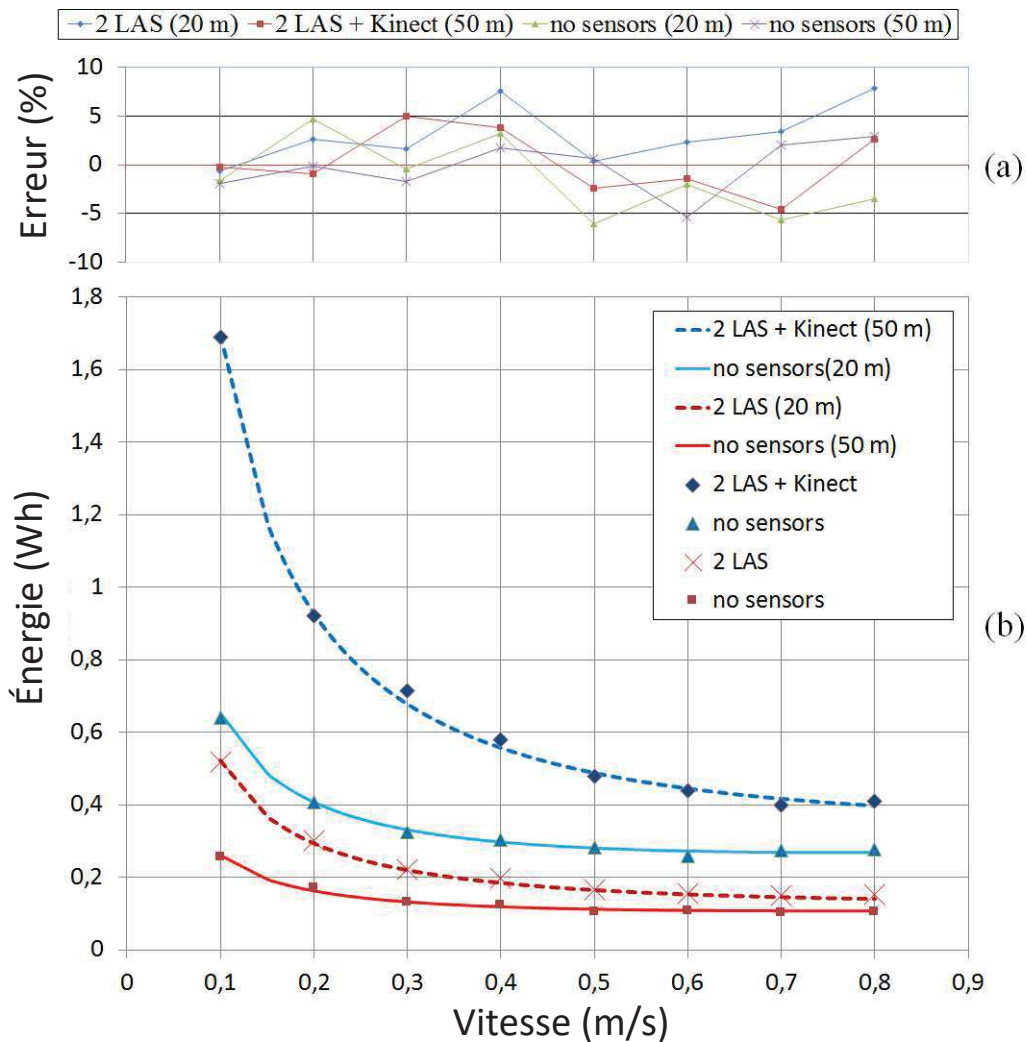


FIGURE 4.15 – Estimation de l'énergie consommée sur la batterie du robot (mission Avance Simple) : (a) Erreur de prédiction, (b) Résultats expérimentaux et théoriques

Tableau 4.7 – Tableau récapitulatif des extremums des erreurs d'estimation de la consommation énergétique sur la batterie du robot, la moyenne et l'écart type

capteurs	distance (m)	erreur			
		min (%)	max (%)	moyenne (%)	écart type (%)
<i>sans capteurs</i>	20	-6.05	+4.67	-1.41	+3.58
	50	-5.32	+2.90	-0.2	+2.50
<i>2 lasers</i>	20	-0.60	+7.84	+3.14	+2.87
<i>2 lasers + kinect</i>	50	-4.5	+4.95	+0.22	+3.05

Pour les différentes AI, l'erreur de prédiction est bornée entre -6% et 8% avec une erreur moyenne comprise entre -1.41% et 3.14%. L'écart type est proche de 3%. Ceci démontre une bonne précision de prédiction validée expérimentalement pour différentes vitesses et pour différentes AI.

c.2. Batterie du PC

De même, la figure 4.16(b) montre les relevés expérimentaux de la consommation sur la batterie du PC pour les différentes AI d'Avance Simple. La figure 4.16(a) et le tableau 4.8 montrent que l'erreur de prédiction varie entre -6.6% et 2.6% avec une erreur moyenne de 2.8%. L'écart type est approximativement de 2.5%.

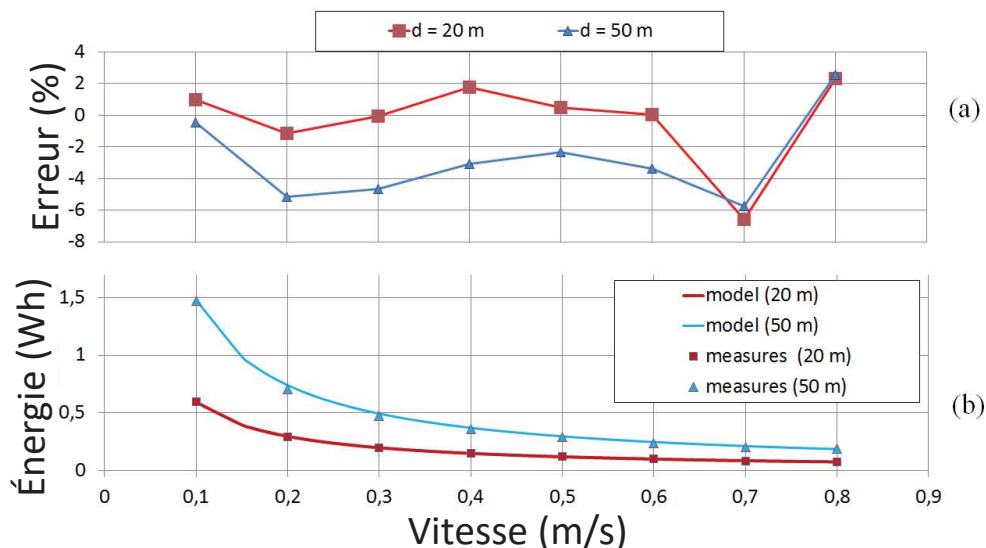


FIGURE 4.16 – Estimation de l'énergie consommée sur la batterie du robot (mission Avance Simple : (a) Erreur de prédiction, (b) résultats expérimentaux et théorique

Tableau 4.8 – Tableau récapitulatif des extremums des erreurs d'estimation de la consommation énergétique sur la batterie du PC, la moyenne et l'écart type

distance (m)	erreur			
	min (%)	max (%)	moyenne (%)	écart type (%)
20	-6.6	+2.32	+0.27	+2.59
50	-4.5	+2.58	+2.77	+2.56

Ces expériences montrent que les modèles d'estimation de la consommation énergétique, que cela soit sur la batterie du robot ou du PC, permettent d'avoir une prédiction de bonne précision. Bien évidemment la dimension énergétique sera fortement influencée par la durée d'exécution des différentes activités de la mission. Nous aborderons dans la suite cet axe de performance.

4.1.3 Axe durée

Après avoir établi les modèles d'estimation de performances sécuritaire et énergétique, nous allons poursuivre maintenant notre étude avec l'axe de durée.

4.1.3.1 Graphe des indicateurs/inducteurs de performance de l'axe durée

Même si les axes précédents sont indispensables pour assurer la réalisation d'une mission, l'axe durée, bien que potentiellement facultatif, reste une caractéristique centrale pour une mission de robotique mobile. Sans cette contrainte, le robot ne bougera pas pour rester le plus en sécurité et minimiser sa consommation énergétique. La considération de la durée totale de la mission est nécessaire pour assurer l'existence d'une solution non triviale au problème d'optimisation.

Le diagramme d'Ishikawa de la figure 4.17 montre les principaux éléments qui impactent la durée d'une mission robotique. De même que pour les axes sécurité et énergie, l'Homme ne peut pas impacter la durée de la mission une fois cette dernière définie. A partir des figures 4.17 et 4.18 représentant le diagramme causes à effet et celui d'interdépendance, nous pouvons remarquer que la durée dépend des méthodes mises en œuvre. D'une part le planificateur génère un chemin à suivre et d'autre part la loi de commande et d'évitement d'obstacle conduit le robot à suivre cette trajectoire. Cependant, pour un même chemin à suivre, la distance réellement parcourue peut différer selon la loi de commande adoptée et ses paramètres.

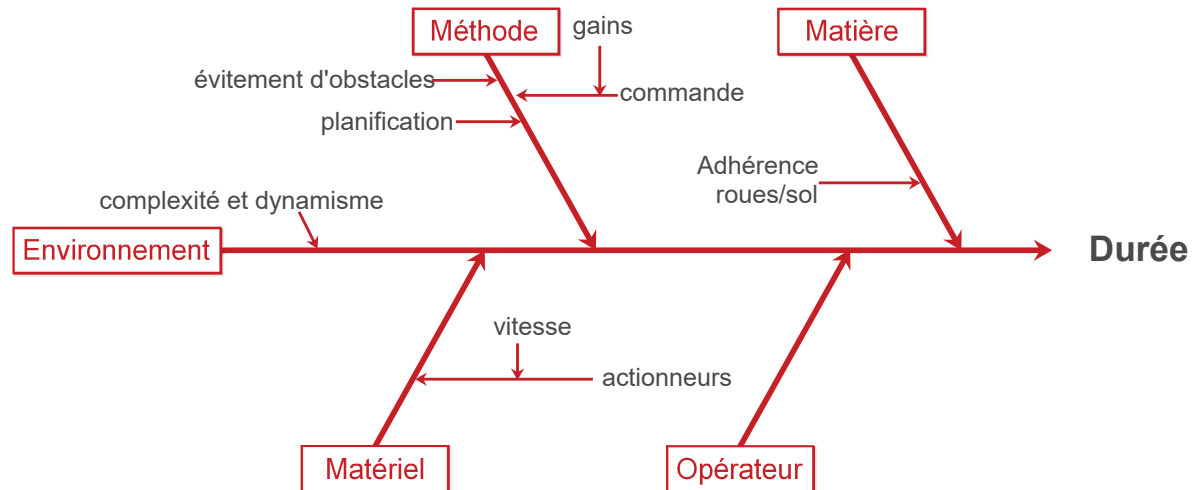


FIGURE 4.17 – Diagramme causes-effets pour l'axe durée appliqué au système d'étude

Ainsi, si différents chemins peuvent être suivis, le choix du planificateur (s'il est utilisé) et des critères qui lui sont associés peuvent impacter considérablement la durée de la mission. Ce cas de figure n'est pas considéré dans le cadre de notre étude. Nous supposons que le chemin choisi est celui proposé par le planificateur. Les chemins générés pour un environnement donné à distances égales peut avoir différentes complexités (virages). Le robot doit alors réduire sa vitesse pour virer et cela peut *a priori* impacter la durée de la mission. Dans notre expérimentation nous supposons cet impact négligeable puisque le robot se déplace souvent en ligne droite. Le dynamisme de l'environnement peut induire aussi d'éventuels évitement d'obstacles et donc un rallongement possible de la durée de la mission.

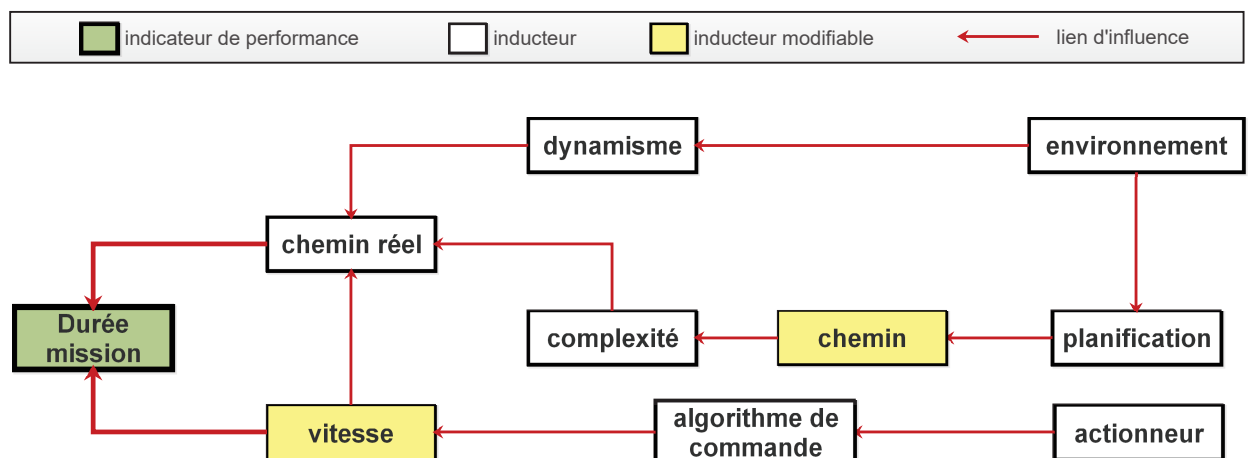


FIGURE 4.18 – Diagramme d'interdépendance pour l'axe durée appliqué au système d'étude

Cet élément sera pris en compte pour nos simulations et expérimentations puisqu'une partie de la mission se fait en présence humaine. Comme nous le verrons cela impactera de façon non négligeable la durée de la mission et donc la consommation énergétique du robot.

Il faut aussi remarquer que le temps nécessaire au robot pour réaliser sa mission ne correspond pas nécessairement à des tâches où le robot effectue un déplacement (de sa base ou d'un organe de manipulation). En effet le robot peut aussi s'arrêter par exemple pour attendre un ordre, pour acquérir des données, pour échanger des informations, pour attendre la fin d'un traitement informatique.

Nous allons maintenant établir un modèle d'estimation de la durée d'une mission robotique pour l'indicateur de performance $Perf_{SR, \text{Durée}}$; Durée.

4.1.3.2 Modèle d'estimation de la durée

a. Travaux existants

La problématique de la durée de mission est rarement abordée dans la littérature. De façon surprenante, son estimation reste liée à beaucoup de facteurs. Certains travaux l'ont abordée comme une contrainte de réalisation de la mission.

Ota dans [Ota09] propose une méthode de recherche d'une séquence d'actions optimale avec optimisation des objectifs sous une contrainte temporelle. L'objectif est de prendre en compte les temps nécessaires pour la planification et celui d'exécution. Il admet que le temps d'exécution d'une tâche est incertain et propose une méthode incluant un coefficient d'erreur qu'il met à jour lors de l'exécution de la mission. Le problème d'optimisation est abordé aussi dans [WV07 ; WV08]. Ils s'intéressent au problème de recharge périodique des robots mobiles autonomes effectuant des missions de longue durée. Ils proposent une méthodologie permettant de maximiser le temps de travail du robot tout en décidant du meilleur moyen (point de recharge, panneau solaire) et l'endroit de recharge.

Heero *et al.* dans [HAK] s'intéressent aux missions d'exploration. Ils proposent une méthodologie permettant d'enrichir la connaissance sur l'environnement exploré. Elle permet d'exploiter la connaissance sur les durées de traversée des cellules de la grille de l'environnement pour améliorer l'efficacité de point de vue du temps d'exploration du reste de la mission.

Dans les précédents travaux on peut remarquer qu'un minimum de connaissance *a priori* sur la mission à réaliser est nécessaire pour gérer la durée d'une mission. Même dans la cas de missions d'exploration, les objectifs sont connus mais le temps nécessaire

pour les atteindre sont acquis au cours de la mission. Nous pouvons aussi remarquer que la vitesse est toujours considérée comme constante par segment de chemin.

Nous allons différencier dans la suite les tâches d'attente de celles de déplacement et de traitement afin de formaliser la durée d'une activité.

b. Proposition

b.1. Durée d'attente

Les missions robotiques peuvent contenir des tâches d'attente ou des actions à durée constante. Ces durées sont prédéfinies par le donneur d'ordre ou préalablement connues pour faire une action (exemple recharge batterie). Cette attente ne peut être exécutée que séquentiellement par rapport aux autres activités. On note les durées d'attente pour une activité i , $D_{att_{A_i}}$.

$$D_{A_i} = D_{att_{A_i}} \quad (4.31)$$

b.2. Durée de déplacement

La durée $D_{dep_{A_i}}$ mise pour effectuer un déplacement est fonction de la distance dis_i à parcourir et de la vitesse $v_{r_{A_i}}$ du robot durant une activité i que nous supposons constante.

$$D_{dep_{A_i}} = \frac{dis_i}{v_{r_{A_i}}} \quad (4.32)$$

b.3. Durée de traitement

Le robot peut effectuer des tâches de traitement, réception ou envoi de données, planification, etc, de durée $D_{trai_{A_i}}$. Ce type de tâche peut être exécuté en parallèle avec avec d'autres activités (déplacement ou autre traitement). La durée totale de l'activité D_{A_i} est le maximum des durées des tâches $n_T^{A_i}$ correspondants aux tâches de cette activité en supposant qu'elles commencent au même temps. La durée d'une activité est exprimée alors comme suit :

$$D_{A_i} = \text{Max}_{j=1}^{j=n_T^{A_i}} (D_{dep_{A_i}}, D_{trai_{A_i}}) \quad (4.33)$$

b.4. Formulation du modèle d'estimation de la durée

La durée totale d'une mission est évidemment équivalente à la somme des durées des activités qui sont par définition séquentielles. Elle est exprimée par la loi suivante :

$$Perf_{SR, \text{ Durée}} = \sum_{i=1}^{n_{Act}} D_{A_i} \quad (4.34)$$

Nous allons passer maintenant aux axes de performance de localisation et de stabilité. Ils sont traités heuristiquement dans cette thèse. Cependant nous évoquons quelques pistes qui nous permettrons de les aborder dans des travaux futurs.

4.1.4 Axe localisation

Le diagramme Ishikawa de la figure 4.19 montre les principaux éléments qui impactent la localisation d'un robot mobile en terme de précision. Nous pouvons constater que cette dernière dépend principalement de :

- Environnement : il est représenté par un modèle. Son impact sur la localisation réside dans la fidélité et la granularité de ce modèle par rapport à l'environnement réel.
- Ressources matérielles : ce sont principalement les capteurs proprioceptifs et extéroceptifs. Elles permettent dans le cas du SLAM de construire une carte de l'environnement dans laquelle le robot se localise. Leur précision ainsi que leur technologie, en relation avec la nature de l'environnement, agissent sur la fiabilité et la fidélité du modèle de l'environnement. Ces ressources peuvent être des instruments de mesures qui sont propres au robot utilisé pour mesurer les positions de balises instrumentant l'environnement (points de repère). Nous pouvons aussi rajouter le CPU qui supporte l'architecture de contrôle et dont la puissance de calcul impacte la précision de localisation en termes de fréquence d'acquisition des données capteurs et la taille de données pouvant être traitées.
- Ressources logicielles : Elles permettent d'interpréter et de remettre en forme les informations provenant des capteurs (fusion des données). Les algorithmes de localisation peuvent donner des informations plus au moins précises. De plus, tous les précédents algorithmes sont exécutés au sein d'une architecture de contrôle exécutée sur un CPU qui a ses propres caractéristiques en termes de fréquences d'exécution et de mémoire.

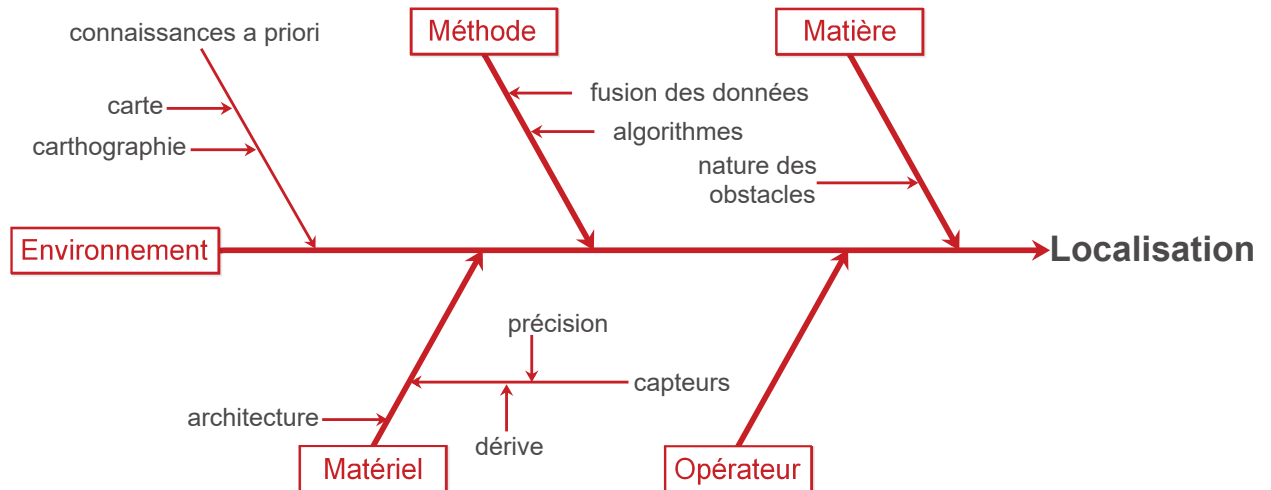


FIGURE 4.19 – Diagramme causes-effets pour l'axe localisation appliqué au système d'étude

Le traitement de cet axe va correspondre par la suite à des contraintes d'utilisation de certains algorithmes avec des paramètres définis. Ces conditions ont été établies expérimentalement.

4.1.5 Axe stabilité

La stabilité est impactée par plusieurs éléments (Figure 4.20). Principalement ce sont les méthodes utilisées (algorithmes). La stabilité est une propriété à assurer pour tout système régulé. Le principe d'une "marge de stabilité" est parfaitement défini pour le cas des systèmes linéaires. Mais les modèles de robots mobiles s'accommodent mal de cette hypothèse qui conduit à linéariser les modèles et perdre la géométrie du problème, voire le rendre non-commandable s'il est non-holonome, comme le montre Brockett [Bro+83]. Ainsi, les commandes des systèmes mobiles sont non-linéaires, et parfois même discontinues (pour une présentation des différentes approches de la commande d'un robot mobile non-holonome – cas de notre étude – se référer à [LSP06] et aux références que ces travaux contiennent). Cette non-linéarité fait que les techniques de conception issues du linéaire ne sont plus applicables. Cependant, la conception de ces commandes non-linéaires suit une approche "par la preuve", et confère à la solution une garantie formelle de convergence du (modèle du) système vers son objectif, ou dans un voisinage borné. Cette garantie mathématique forte doit pouvoir être exploitée. La transposition du principe de marge au cas non-linéaire nécessite de difficiles manipulations algébriques, et reste une question encore ouverte.

La stabilité concerne aussi la question de la navigation (localisation). Le "filtre de fu-

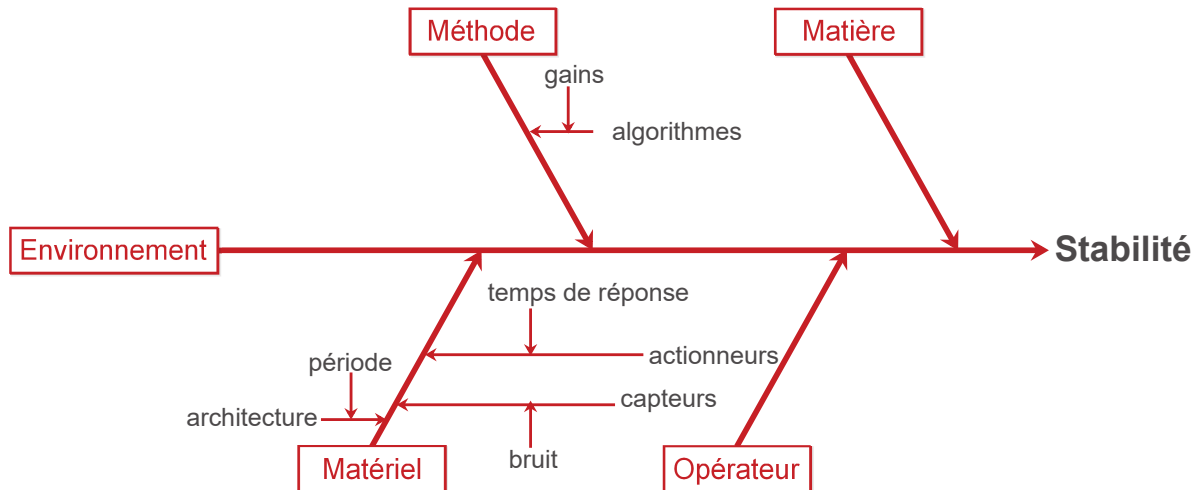


FIGURE 4.20 – Diagramme causes-effets pour l'axe stabilité appliqué au système d'étude

sion de données" qui, d'après les mesures capteurs, value les entrées nécessaires au calcul du contrôle, et pour lequel la stabilité doit aussi être assurée. La considération des bruits capteurs peut être approchée par des méthodes stochastiques, comme les filtres de Kalman étendu ou particulière, mais à nouveau la non-linéarité du problème ne permet pas d'obtenir de garanties formelles (et globales) de convergence du filtre. Cependant l'approche des filtres complémentaires permet, par des techniques identiques à celles employées pour le contrôle, d'obtenir une telle garantie (BIBO) [KPK99].

Il est un aspect essentiel du système qui n'est que marginalement abordé dans la littérature : la question de l'échantillonnage. L'effet de discrétisation de la mesure sur le filtre (navigation) peut être explicitement considéré avec les approches non-linéaires des systèmes commutés [HM99]. Mais la prise en compte de cet effet sur le contrôle, et de son impact sur les propriétés de convergence du système fait actuellement encore l'objet d'intenses recherches.

L'effet de l'échantillonnage doit aussi être considéré à l'exécution de la commande. Ceci concerne le choix de la période d'échantillonnage du contrôle, une question cruciale dont la résolution dépasse le cadre de nos travaux, mais sur lesquels elle a un impact considérable. En effet, la modification de la période d'échantillonnage du contrôle impacte l'ensemble du système, et les 4 autres axes de performance. D. Simon aborde cette question dans [SSA13], par l'analyse de la "gigue" sur des systèmes dynamiques. Le domaine de l'*autonomic computing* traite aussi de ce sujet, mais considère des systèmes de calculateurs, sans contraintes physiques. Le domaine naissant des Systèmes Cyber-Physiques (SCP) cible directement ce type de problème.

Dans ces travaux, nous avons suivi l'approche classique de roboticien qui consiste à surdimensionner arbitrairement la période d'échantillonnage du contrôle. Comme tant d'autres, nous avons choisi $T_e = 0.1$ s, ce qui semble "raisonnable" au vu de la dynamique du système. Ceci se vérifie par la linéarisation, la simulation et bien sûr expérimentalement.

La première étape de la phase préliminaire P1 de l'approche proposée concernant l'identification des modèles d'estimation de performance pour les différents axes retenus est maintenant achevée. Nous passons maintenant à l'étape d'identification du scénario de mission détaillé nécessaire pour le calcul d'une solution d'allocation de ressources lors de la deuxième phase P2.

4.1.6 Identification du Scénario de Mission Détaillé (SMD)

Nous allons établir dans cette partie le SMD à partir de la description de mission, d'un point de vue utilisateur (Table 3.2). Les différentes contraintes de performance et environnementales vont être projetées sur la mission tout en précisant les limites spatio-temporelles de chacune des activités ainsi que les différentes AI éligibles correspondantes.

Le tableau 4.9 récapitule les différentes AI qui correspondent aux différentes tâches pour les objectifs de la mission cible. Les objectifs de déplacement d'un point à un autre (O_I , O_V et O_{IX}), comme le montre le tableau 3.2, nécessitent l'exécution de deux tâches en parallèles (Déplacement Vers (**DV**) et Localisation (**LN**)). Comme le déplacement peut utiliser une des 7 options d'implémentation de tâche (OIT) possibles et la localisation un des 3 OIT envisageables (Table 3.1), il en résulte que nous avons 21 façons de nous déplacer. Le même principe de raisonnement s'applique pour les objectifs permettant au robot de tourner sur place (Tourner Vers (**TV**)) pour les objectifs O_{II} , O_{IV} , O_{VI} et O_{VIII} et d'inspecter la vanne (Analyse d'Image (**AEIG**)) pour les objectifs O_{III} et O_{VII} .

Tableau 4.9 – Tableau de composition des différentes AI et leurs références correspondantes

Tâche	DV LN	Ref.	TV LN	Ref.	AEIG	Ref.		
AI	<i>SC-LOD</i>	1	<i>TV-LOD</i>	22	DV	25		
	<i>S_US-LOD</i>	2	<i>TV-LQR</i>	23				
	<i>S_L-LOD</i>	3	<i>TV-LCH</i>	24				
	<i>S_2LUS-LOD</i>	4						
	<i>S_2L-LOD</i>	5						
	<i>C_2L-LOD</i>	6						
	<i>C_2LUS-LOD</i>	7						
	<i>SC-LQR</i>	8						
	<i>S_US-LQR</i>	9						
	<i>S_L-LQR</i>	10						
	<i>S_2LUS-LQR</i>	11						
	<i>S_2L-LQR</i>	12						
	<i>C_2L-LQR</i>	13						
	<i>C_2LUS-LQR</i>	14						
	<i>SC-LCH</i>	15						
	<i>S_US-LCH</i>	16						
	<i>S_L-LCH</i>	17						
	<i>S_2LUS-LCH</i>	18						
	<i>S_2L-LCH</i>	19						
	<i>C_2L-LCH</i>	20						
	<i>C_2LUS-LCH</i>	21						

4.1.6.1 SMD sans prise en compte des contraintes de performance

Comme l'indique la méthodologie de gestion proposée et l'illustre la figure 2.8, le SMD est établi en fonction des contraintes de performances booléennes (sécurité et localisation) prises en compte. En premier lieu le SMD est construit sans prise en compte des contraintes environnementales ou de performance. La description du SMD est récapitulée dans le tableau 4.11 où nous trouvons en première ligne les différents objectifs de la mission cible ainsi que leurs coordonnées linéaires de début et de fin (x_d et x_f). Les tâches nécessaire pour réaliser les différents objectifs sont dans la deuxième ligne suivies par les la décomposition en activités dans la troisième ligne du tableau. Nous trouvons ensuite le vecteur d'information I contenant la coordonnée linéaire de début x_{A_i} de l'activité i , sa distance linéaire dis_{A_i} ainsi que sa durée dur_{A_i} . Enfin, l'ensemble des AI éligibles par

activité (Liste des AI) représente l'ensemble global de configurations du tableau 4.9.

On remarque que les activités correspondent exactement aux objectifs et aucune modification n'a été apportée sur la réalisation de la mission puisque aucun axe de performance booléen n'a été pris en compte. Ceci ne sera pas le cas s'il y avait, par exemple, des contraintes temporelles sur des parties/zones de la mission (e.g. parcourir un couloir sous une durée déterminée). Les coordonnées linéaires des activités (vecteur I du Tableau 4.11) correspondent à ceux des objectifs de la mission. Même si les activités de rotation vers la vanne ou vers le chemin à poursuivre (A_2 , A_4 , A_6 et A_8) consiste en une rotation du robot sur lui même, elles ne permettent pas une progression spatiale dans la mission. De plus, leur durée d'exécution sont prédéfinies. C'est aussi le cas pour les activités de détection de l'état des vannes par analyse d'image *AEIG*.

Aucune restriction de vitesse n'est appliquée sur les différentes activités puisque aucun axes de performance booléen n'est pris en compte. Toutes les activités de déplacement ont une vitesse maximale qui correspond à la vitesse maximale du robot (couleur orangée, voir les codes de couleur du tableau 4.10). Les activités dont les durées sont prédéfinies et sans déplacement linéaire sont en gris (0 m/s).

Tableau 4.10 – Code de couleur des vitesses maximales par zone

Vitesse limite	Code de couleur
0	
$v_{US_{max}}$	
v_{choc}	
$v_{r_{max}}$	

Cet exemple montre que si aucun axe de performance n'est pris en compte pour établir un SMD, le nombre d'activités est identique à celui des objectifs. De plus, toutes les combinaisons de configurations possibles pour les objectifs restent éligibles pour les activités.

Tableau 4.11 – S  nario de Mission D  taill   sans contraintes

Objectif ($x_d - x_f$)		O_1 (0 - 37)	O_2 (37)	O_3 (37)	O_4 (37)	O_5 (37 - 93.5)	O_6 (93.5)	O_7 (93.5)	O_8 (93.5)	O_9 (93.5 - 187)
T��che		DV LN	TV LN	AEIG	TV LN	DV LN	TV LN	AEIG	TV LN	VD LN
Activit��		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
I	x_{A_i}	0	37	37	37	37	93.5	93.5	93.5	93.5
	dis_{A_i}	37	0	0	0	56.5	0	0	0	93.5
	dur_{A_i}	-	1	10	1	-	10	1	10	-
Liste des AI (Table 4.9)		1				1				1
		2				2				2
		3				3				3
		4				4				4
		5				5				5
		6				6				6
		7				7				7
		8				8				8
		9				9				9
		10	22		22	10	22		22	10
		11	23	25	23	11	23	25	23	11
		12	24		24	12	24		24	12
		13				13				13
		14				14				14
		15				15				15
		16				16				16
		17				17				17
		18				18				18
		19				19				19
		20				20				20
		21				21				21

4.1.6.2 SMD avec prise en compte de l'axe sécurité

Le SMD avec prise en compte des contraintes liées à la sécurité est montré sans le tableau 4.12. Comme pour le SMD établi précédemment, ce tableau récapitule les informations (vecteur I) sur les activités ainsi que les AI éligibles correspondantes.

La prise en compte des différentes contraintes c_1 (zone de présence humaine), c_2 (zone sans présence humaine), c_3 (zone de vitres) liées à la sécurité (section 3.2.3) implique un découpage des objectifs de déplacement (O_1 , O_5 et O_9) en activités à contraintes invariantes. L'objectif O_1 s'étend en premier lieu sur une zone de présence humaine sous la contrainte c_1 . Toutes les AI possibles restent utilisables pour A_1 . Cependant, la vitesse est réduite à v_{choc} (couleur verte) pour assurer la contrainte sécuritaire d'innocuité. Le même raisonnement s'applique pour les activités A_7 , A_{13} et A_{15} .

Le robot passe ensuite dans une zone de vitre sous la contrainte c_3 (Figure 3.18). La contrainte sécuritaire d'évitement d'obstacle dans cette zone implique la réduction du nombre d'AI éligibles pour A_2 . Toutes les AI ne mobilisant pas les sonars ne pourront alors pas être utilisées (Table 4.9). Puisque les seuls capteurs de proximités efficaces dans cette zone sont les sonars, la vitesse est limitée à $v_{US_{max}}$. Ailleurs, si uniquement c_1 est imposée, la vitesse est limitée à v_{choc} (couleur verte).

Le déplacement du robot entre la vanne V_1 et V_2 puis de V_2 vers la position de départ implique un passage par le couloir CR_2 . La contrainte c_2 qui s'applique sur ce couloir indique la non présence humaine dans cette zone. Ainsi la contrainte d'innocuité imposée par l'axe sécuritaire n'est plus nécessaire et le robot peut rouler jusqu'à sa vitesse maximale (A_8 et A_{12}).

La prise en compte de l'axe sécuritaire a donc impliqué le passage d'une description de la mission du point de vue utilisateur initialement composée de 9 objectifs en une séquence de 15 activités correspondants chacune à une (des) zone(s) de contrainte(s) invariante(s). Pour chaque activité, une vitesse adaptée aux contraintes imposées par la zone traversée a été choisie.

Tableau 4.12 – SMD avec prise en compte de l'axe sécurité

Objectif (x_d - x_f)		O_1 (0 -37)		O_2 (37)	O_3 (37)	O_4 (37)	O_5 (37 - 93.5)			O_6 (93.5)	O_7 (93.5)	O_8 (93.5)	O_9 (93.5 - 187)			
Tâche		DV LN		TV LN	AEIG	TV LN	DV LN			TV LN	AEIG	TV LN	DV LN			
Activité		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
I	x_{A_i}	0	31	37	37	37	37	41	63.5	93.5	93.5	93.5	93.5	123.5	146	156
	dis_{A_i}	31	6	0	0	0	4	22.5	30	0	0	0	30	22.5	10	31
	dur_{A_i}	-	-	10	1	10	-	-	-	10	1	10	-	-	-	-
Liste des AI (Table 4.9)		1	1	22	25	22	1	1	1	22	25	22	1	1	11	1
		2						2	2				2			
		3						3	3				3			
		4						4	4				4			
		5						5	5				5			
		6						6	6				6			
		7						7	7				7			
		8						8	8				8			
		9						9	9				9			
		10						10	10				10			
		11						11	11				11			
		12						12	12				12			
		13						13	13				13			
		14						14	14				14			
		15						15	15				15			
		16						16	16				16			
		17						17	17				17			
		18						18	18				18			
		19						19	19				19			
		20						20	20				20			
		21						21	21				21			

4.1.6.3 SMD avec prise en compte des contraintes de performance sécurité et localisation

En plus des contraintes c_1 , c_2 et c_3 prises en compte précédemment pour la construction du *SMD*, nous rajoutons maintenant celle liée à la localisation c_4 (voir section 3.2.3). Elle concerne la précision de la localisation à l'approche des vannes.

La prise en compte de toutes ces contraintes implique alors un découpage de certains objectifs en activités à contraintes invariantes. L'objectif O_1 qui consiste à aller du point de départ jusqu'à la vanne V_1 se recouvre en premier temps avec la zone de vitre (c_3) et en deuxième temps avec la zone d'approche de la vanne (c_4) tout en restant dans la zone vitrée (Figure 3.18). De plus, c'est une zone de présence humaine (c_1). Ainsi, avant d'atteindre les zones de contraintes c_3 et c_4 , l'activité A_1 reste réalisable avec toutes les AI possibles pour la tâche de déplacement *DV* et localisation *LN* (Table 4.9). Cependant, la contrainte c_1 implique une réduction de la vitesse à v_{choc} (couleur verte). Les activités A_2 et A_3 sont toutes les deux dans la zone de vitres (c_3). Seules les AI impliquant l'utilisation des sonars sont éligibles. La vitesse est ainsi réduite à $v_{US_{max}}$ ($\min(v_{US_{max}}, v_{choc})$) (Figure 4.6)) et l'ensemble d'AI est réduit à 12 pour l'activité A_2 ne gardant alors que celles mobilisant les sonars. Quand à l'activité A_3 , l'utilisation des AI avec localisation basé QR-codes est imposée par c_4 . Ainsi, seulement 4 AI vérifient les contraintes c_3 et c_4 .

Dès que le robot atteint la première vanne, trois activités statiques (position linéaire fixe à 37 m) s'enchaînent. L'ensemble d'AI possibles pour la rotation avec localisation (A_4 et A_6) ou d'analyse d'image (A_5) (Table 4.9) est identique à la liste des AI éligibles puisque le robot est linéairement statique et qu'aucune des 4 contraintes n'est valide.

Encore dans la zone de vitre, l'objectif O_5 est divisé en 4 activités. Elle consiste en un déplacement de la vanne $V1$ jusqu'à $V2$. Le robot doit parcourir différentes zones (deux couloirs CR_1 et CR_2) avec différentes contraintes (vitre dans CR_1 et approche de la vanne $V2$ dans CR_2). La première correspond à A_7 une zone de vitre et de présence humaine où seules les AI avec des sonars sont éligibles. Sur la seconde A_8 , seule la contrainte de présence humaine s'applique et les 21 configurations possibles d'AI restent éligibles mais avec une vitesse limitée à v_{choc} . Dans A_8 le robot rentre dans une zone de non présence humaine (contrainte c_2 dans CR_2). Ainsi, la contrainte de limitation de vitesse est levée et le robot peut rouler jusqu'à sa vitesse maximale $v_{r_{max}}$. De même que pour la vanne $V1$, à l'approche de $V2$ la contrainte c_4 réduit l'ensemble des AI éligibles à celles utilisant les QR-codes pour la localisation. Par conséquent, 7 AI restent utilisables dans l'activité A_{10} .

Tableau 4.13 – Scénario de Mission Détaillé avec prise en compte des axes de performance sécurité et localisation

Objectif (x_d - x_f)		O_1 (0 -37)			O_2 (37)	O_3 (37)	O_4 (37)	O_5 (37 - 93.5)				O_6 (93.5)	O_7 (93.5)	O_8 (93.5)	O_9 (93.5 - 187)			
Tâche		DV LN			TV LN	AEIG	TV LN	DV LN				TV LN	AEIG	TV LN	DV LN			
Activité		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
I	x_{A_i}	0	31	34	37	37	37	37	41	63.5	91.5	93.5	93.5	93.5	93.5	123.5	146	156
	dis_{A_i}	31	3	3	0	0	0	4	22.5	28	2	0	0	0	30	22.5	10	31
	dur_{A_i}	-	-	-	10	1	10	-	-	-	-	10	1	10	-	-	-	-
Liste des AI (Table 4.9)		1							1	1					1	1		1
		2							2	2					2	2		2
		3							3	3					3	3		3
		4							4	4					4	4		4
		5	1					1	5	5					5	5	1	5
		6	2					2	6	6					6	6	2	6
		7	4					4	7	7					7	7	4	7
		8	7					7	8	8	8				8	8	7	8
		9	8	8				8	9	9	9				9	9	8	9
		10	9	9	22		22	9	10	10	10	22		22	10	10	9	10
		11	11	11	23	25	23	11	11	11	11	23	25	23	11	11	11	11
		12	14	14	24		24	12	12	12	12	24		24	12	12	14	12
		13	15					13	13	13					13	13	15	13
		14	16					14	14	14	14				14	14	16	14
		15	18					15	15						15	15	18	15
		16	21					16	16						16	16	21	16
		17						17	17						17	17		17
		18						18	18						18	18		18
		19						19	19						19	19		19
		20						20	20						20	20		20
		21						21	21						21	21		21

Les activités permettant de s'orienter vers la vanne V_2 et de l'inspecter A_{11} , A_{12} et A_{13} se déroulent identiquement à celles permettant d'inspecter V_1 .

Le retour vers le point de départ doit être assuré par la suite (dernier objectif O_9). Cet objectif se réalise avec 4 activités puisqu'il s'agit d'un passage par la zone de non présence humaine (CR_2) en premier lieu et la zone de présence humaine dans CR_1 contenant elle même une zone de vitres.

4.1.7 Conclusion

À l'issu de la première phase de l'approche de gestion des ressources matérielles et logicielles, nous avons établi des modèles d'estimation de la performance selon les axes sécuritaire, énergétique et de durée après avoir identifier les différents inducteurs de performance. Nous avons construit par la suite différents SMD en projetant les axes de performance booléens de sécurité et de localisation un par un. Cela a permis de montrer l'impact de la prise en compte de ces axes sur les limitations de vitesse ainsi que les différents AI éligibles par activité.

Nous allons entamer maintenant la deuxième phase de l'approche proposée. Elle consiste à calculer une solution initiale et hors ligne d'allocation de ressources pour le SMD qui vient d'être construit.

4.2 Méthodologie : Phase hors ligne P2

La deuxième phase de la méthodologie d'affectation des ressources matérielles et logicielles consiste en une estimation de la faisabilité de la mission. Cette estimation est basée sur la description du SMD que nous venons de construire. Si la mission est réalisable, alors une solution d'allocation de ressources *SAR* doit être calculée.

Nous allons prendre en compte les SMD établis sans prise en compte des axes de performance booléens, puis en ne considérant que l'axe sécurité et enfin nous nous intéresserons aux axes sécurité et localisation. Cela nous permettra de montrer la différence entre les solutions trouvées. Les axes de durée et énergie correspondants quant à eux à des performances continues sont pris en compte dès à présent (la stabilité a été traitée arbitrairement).

Sans perdre de généralité, pour décrire la démarche de la méthodologie, nous choisissons les contraintes de performances de durée et énergie suivantes :

- **Axe durée :** La contrainte temporelle est fixée à $Perf_{C, Duree} = 600$ s. Ainsi $Perf_{SR, Duree} \leq Perf_{C, Duree}$ pour que la mission soit réalisable du point de vue durée.
- **Axe énergie :**
 - Consommation maximale sur la batterie du robot est $Perf_{C, Energie}^{Robot} = 2.5$ Wh.
 - Consommation maximale sur la batterie de l'ordinateur portable est $Perf_{C, Energie}^{PC} = 2.8$ Wh.

Pour que la mission soit réalisable du point de vue énergétique, $Perf_{SR, Energie}^{Robot}$ doit être $\leq Perf_{C, Energie}^{Robot}$ et $Perf_{SR, Energie}^{PC} \leq Perf_{C, Energie}^{PC}$.

Il faut rappeler que la prise en compte de l'axe stabilité se fait "empiriquement" en fixant la fréquence d'exécution du contrôle à 10 Hz. Cette fréquence assure expérimentalement la stabilité de la commande du robot.

4.2.1 Mise en œuvre de la phase P2 sur SMD sans performances booléennes et avec prise en compte de l'axe durée

L'analyse de la faisabilité de la mission sans prise en compte des axes de performance booléens est basée sur la description du SMD du tableau 4.11 qu'on note SMD_{sc} (sc : sans contraintes). La complexité du problème se calcule à partir de (2.8), et est exprimée pour le cas traité maintenant dans (4.35). Le nombre d'AI de chaque activité est récapitulé dans le tableau 4.14. Nous trouvons aussi les vitesses maximales $v_{A_{i_{max}}}$ et minimale $v_{A_{i_{min}}}$, les durées $dur_{A_{j_{min}}}$ et les distances linéaires dis_{A_i} de chaque activité dis_{A_i} .

$$NGA_{SMD_{sc}} = \prod_{i=1}^9 n_{AI}^{A_i} = 750141 \quad (4.35)$$

Tableau 4.14 – Récapitulatif du SMD sans prise en compte des axes de performance booléens et prise en compte de l'axe durée

Activité A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
$n_{AI}^{A_i}$	21	3	1	3	21	3	1	3	21
$v_{A_{i_{max}}} \text{ (m/s)}$	0.76	-	-	-	0.76	-	-	-	0.76
dis_{A_i}	37	0	0	0	56.5	0	0	0	93.5
$dur_{A_{j_{min}}} \text{ (s)}$	48.68	10	1	10	74.34	10	1	10	123.02
$v_{A_{i_{min}}} \text{ (m/s)}$	0.10	-	-	-	0.14	-	-	-	0.21

L'étude de faisabilité du point de vue durée nécessite le calcul de la vitesse minimale par activité pour la contrainte $Perf_{C, \text{Duree}}$ considérée. Si la vitesse minimale par activité ne dépasse pas la vitesse maximale permise pour l'activité correspondante la mission est alors réalisable de point de vue durée. La vitesse $v_{A_{i\min}}$ pour l'activité A_i est calculée par (4.36). Pour l'activité A_i , cette vitesse permet de réaliser la mission avec la durée $Perf_{C, \text{Duree}}$ tout en roulant à la vitesse maximale pour les autres activités.

$$v_{A_{i\min}} = \frac{dis_{A_i}}{dur_{A_{imax}}} = \frac{dis_{A_i}}{Perf_{C, \text{Duree}} - \sum_{j=1/j \neq i}^{n_{act}} dur_{A_{j\min}}} \quad (4.36)$$

Avec :

$$\begin{cases} si & dis_{A_j} \neq 0 & alors & dur_{A_{j\min}} = \frac{dis_{A_j}}{v_{A_{j\max}}} \\ sinon & & alors & dur_{A_{j\min}} = dur_{A_j} \end{cases} \quad (4.37)$$

Les durées et vitesses minimales par activité $v_{A_{i\min}}$ sont récapitulées dans le tableau 4.14. Nous pouvons en déduire que la mission est réalisable puisque toutes les vitesses minimales sont inférieures à leurs vitesses maximales correspondantes.

Si la vitesse maximale est retenue partout, la marge de performance (voir section 2.1.3) selon l'indicateur de durée sera estimée comme suit :

$$M_{Duree_{sc}} = Perf_{C, \text{Duree}} - \widehat{Perf_{SR, \text{Duree}_S}} = 312 \text{ s} \quad (4.38)$$

Cependant le choix d'une alternative ne peut être fait d'une façon objective. Toutes les AI permettent théoriquement de réaliser la mission du point de vue durée, du moment où elles sont paramétrées avec les vitesses calculées $v_{A_{imax}}$. Évidemment le traitement de cet axe ne peut être fait que théoriquement car en pratique le robot ne peut assurer sa mission s'il roule à sa vitesse maximale. Il présenterait un danger considérable pour les humains présents sur sa trajectoire. Heureusement, la contrainte sécuritaire sera prise en compte dans la suite.

4.2.2 Mise en œuvre de la phase P2 sur SMD avec prise en compte de l'axe sécurité (booléen) et de l'axe durée (continu)

Le SMD établi précédemment (Table 4.12) est considéré. Le tableau 4.15 récapitule les vitesses maximales et les distances à parcourir lors de l'exécution de chacune des activités de la mission. Les vitesses minimales sont calculées comme nous l'avons fait précédemment avec (4.36).

Tableau 4.15 – Récapitulatif du SMD avec prise en compte de l'axe sécurité

Activité A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
$n_{AI}^{A_i}$	21	12	3	1	3	12	21	21
$v_{A_{i_{max}}} (m/s)$	0.40	0.32	-	-	-	0.32	0.40	0.53
dis_{A_i}	31	6	0	0	0	4	22.5	30
$dur_{A_{j_{min}}} (s)$	77.50	18.75	10	1	10	12.5	56.25	56.60
$v_{A_{i_{min}}} (m/s)$	0.16	0.04	-	-	-	0.03	0.13	0.17

Activité A_i	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
$n_{AI}^{A_i}$	3	1	3	21	21	12	21
$v_{A_{i_{max}}} (m/s)$	-	-	-	0.53	0.40	0.32	0.40
dis_{A_i}	0	0	0	30	22.5	10	31
$dur_{A_{j_{min}}} (s)$	10	1	10	56.60	56.25	31.25	77.5
$v_{A_{i_{min}}} (m/s)$	-	-	-	0.17	0.13	0.06	0.16

La complexité du problème NGA_{SMD_S} (S : Sécurité) calculée avec (2.8) s'élève, pour cet exemple, à plus de $1.2 \cdot 10^{13}$ (4.39).

$$NGA_{SMD_S} = \prod_{i=1}^{15} n_{AI}^{A_i} = 12\ 004\ 512\ 424\ 128 \quad (4.39)$$

Nous pouvons déduire ainsi que la mission reste réalisable puisque toutes les vitesses minimales sont inférieures aux vitesses maximales correspondantes. La vitesse maximale est alors retenue pour toutes les activités et la marge de performance selon l'indicateur de durée sera estimée par (4.40). La marge de durée avec prise en compte de la sécurité est inférieure à celle calculée sans prise en compte des contraintes booléennes. Ceci est expliqué par la réduction de la vitesse maximale des différentes activités. Ainsi, la mission reste théoriquement possible.

$$M_{Duree_S} = Perf_{C, Duree} - \widehat{Perf_{SR, Duree_S}} = 115\ s \quad (4.40)$$

Pratiquement le robot peut réaliser l'aller retour sans présenter de danger pour lui ou son environnement. Cependant, le positionnement précis du robot devant la vanne ne peut être garanti si une AI différentes de celles imposées par la contrainte de localisation c_4 est choisie à l'approche des vannes (activités A_2 et A_8). Le choix ne peut être fait encore une fois objectivement et la discrimination entre les différents AI ne peut être faite. Pour cela, nous allons prendre en compte l'axe de localisation et l'axe énergétique dans la suite afin de trouver une solution d'allocation de ressources SAR qui garantit la réalisation de

la mission.

4.2.3 Mise en œuvre de la phase P2 sur SMD avec prise en compte des axes booléens et continus

Le SMD établi sous les contraintes sécuritaire et de localisation est maintenant considéré (Table 4.13). La détermination des vitesses minimales permet de vérifier que la mission reste réalisable du point de vue durée avec la même marge de durée calculée précédemment (les vitesses maximales sont les mêmes).

A partir du nombre d'AI $n_{AI}^{A_i}$, la complexité du problème d'affectation NGA_{SMD} avec prise en compte de tous les axes de performance booléens peut être alors calculée (4.41). Le nombre de choix possible devient supérieur à $3.3 \cdot 10^{14}$ combinaisons d'AI possibles.

$$NGA_{SMD} = \prod_{i=1}^{17} n_{AI}^{A_i} = 336\ 126\ 347\ 875\ 584 \quad (4.41)$$

Tableau 4.16 – Récapitulatif du SMD avec prise en compte de tous les axes de performance booléens

Activité A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
$n_{AI}^{A_i}$	21	12	4	3	1	3	12	21	21
$v_{A_{i_{max}}} \text{ (m/s)}$	0.40	0.32	0.32	-	-	-	0.32	0.40	0.40
dis_{A_i}	31	3	3	0	0	0	4	22.5	30
$dur_{A_{j_{min}}} \text{ (s)}$	77.50	9.37	9.37	10	1	10	12.50	56.25	75
$v_{A_{i_{min}}} \text{ (m/s)}$	0.16	0.01	0.01	-	-	-	0.02	0.08	0.08

Activité A_i	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
$n_{AI}^{A_i}$	7	3	1	3	21	21	12	21
$v_{A_{i_{max}}} \text{ (m/s)}$	0.53	-	-	-	0.53	0.40	0.32	0.40
dis_{A_i}	2	0	0	0	30	22.5	10	31
$dur_{A_{j_{min}}} \text{ (s)}$	3.77	10	1	10	56.60	56.25	31.25	77.5
$v_{A_{i_{min}}} \text{ (m/s)}$	0.01	-	-	-	0.07	0.05	0.02	0.05

La prise en compte de l'axe énergétique nous permet de discriminer les différentes AI de chaque activité. Ainsi on associe à chaque AI :

- La durée : calculée en fonction de la distance linéaire de l'activité dis_{A_i} et de la vitesse maximale $v_{A_{i_{max}}}$ (voir section 4.1.3.2).

- Un coût énergétique sur la batterie du PC et un autre sur la batterie du robot : calculé en fonction de la configuration matérielle et logicielle de chaque AI (section 4.1.2.2).

L'algorithme de décision sac à dos proposé peut être maintenant utilisé (voir section 2.2.3.2) pour calculer l'ensemble de solutions vérifiant la contrainte de performance énergétique selon l'indicateur $Perf_{C, Energie}^{PC}$. Le sous-ensemble calculé est utilisé comme entrée de l'algorithme pour définir de la même façon un sous-ensemble résultant de la contrainte temporelle $Perf_{C, Duree}$ puis celle de la batterie du robot $Perf_{C, Energie}^{Robot}$.

La stratégie retenue consiste à adopter parmi les solutions permettant de réaliser la mission celle la plus énergivore. Elle permet d'exploiter les AI mobilisant le plus de ressources et qui sont généralement les plus performantes. La solution d'allocation de ressources initiale hors ligne retenue SAR_0 est définie dans le tableau 4.17. Elle a été calculée en 477 itérations. Chaque case de la deuxième ligne (SAR_0) du tableau correspond à une AI dont les références sont détaillées dans la Table 4.9 (11 : S_2LUS_LQR, 23 : TV_LQR, etc.).

 Tableau 4.17 – Détails de la SAR_0 calculée hors ligne

Activité A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
SAR_0	11	11	11	23	25	23	11	11	11	11	23
$dur_{A_{jmin}}$ (s)	78.28	9.31	6.21	10	1	10	15.52	56.82	66.66	4.76	10
$E_{A_j^R}$ (Wh)	0.35	0.03	0.02	0.04	0.02	0.04	0.06	0.25	0.30	0.02	0.04
$E_{A_j^{PC}}$ (Wh)	0.40	0.04	0.03	0.03	0.01	0.03	0.08	0.29	0.34	0.02	0.03
Activité A_i	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	Perf. estimée		Marges		
SAR_0	25	23	11	11	11	1	517		83		
$dur_{A_{jmin}}$ (s)	1	10	71.42	56.82	31.05	78.28	2.28		0.22		
$E_{A_j^R}$ (Wh)	0.02	0.04	0.32	0.25	0.13	0.35	2.59		0.21		
$E_{A_j^{PC}}$ (Wh)	0.01	0.03	0.35	0.28	0.16	0.40					

4.3 Conclusion

Ce chapitre a permis de mettre en œuvre, dans le cadre d'une mission de patrouille, les deux premières phases statiques de la méthodologie de gestion de ressources proposée. Pour cela nous avons identifié, par analyse graphique, lors de la phase préliminaire, les différents indicateurs et inducteurs de performance du système robotique considéré et selon les différents points de vue de la performance. Ensuite des modèles permettant d'estimer la performance pour chaque indicateur de performance ont été proposés. À l'issue de cette

étape, nous avons identifié des scénarii de mission détaillés SMD avec prise en compte des axes de performances booléens et pour différentes contraintes (sans contraintes, sécurité et sécurité/localisation).

La deuxième phase de la méthodologie a ensuite été abordée. Elle consiste à estimer la faisabilité de la mission et à calculer une solution d'allocation de ressources matérielles et logicielles SAR parmi un espace de solutions possibles colossal. Ce travail a été réalisé pour les différents exemples de SMD établis précédemment. Cette étape a montré que hors du contexte théorique, la mission n'est réalisable que si l'on prend en compte tous les axes de performance. De plus, l'axe énergétique apporte un critère objectif de choix de l'alternative d'implémentation AI permettant de réaliser chaque activité de la mission.

Maintenant que la mission est estimée réalisable et qu'une solution d'allocation de ressources SAR_0 a été calculée, la troisième et dernière phase de la méthodologie peut être abordée. Durant celle-ci, le robot exécute la mission avec un suivi de la performance pratiquement mesurée par rapport à celle estimée. La faisabilité de la mission est vérifiée en ligne et en temps réel par rapport aux dérives de performance et aux potentiels fautes sur les ressources utilisées. En cas d'impossibilité de réalisation de la mission avec la solution courante, une nouvelle solution est calculée en ligne (si elle existe). Cette phase va être présentée et validée par simulation et sur une mission réelle dans le prochain chapitre.

Chapitre 5

Mise en œuvre de la phase dynamique de la méthodologie

Sommaire

5.1 Simulation	163
5.1.1 Description de l'outil de simulation	163
5.1.2 Simulation	164
5.2 Expérimentation	173
5.2.1 Conditions expérimentales	173
5.2.2 Déroulement de la mission	173
5.3 Conclusion	179

Après avoir présenté dans le chapitre précédent les deux premières phases hors ligne de l'approche de gestion des ressources proposée, nous allons détailler dans ce chapitre la mise en œuvre de la phase en ligne. La mise en œuvre de la troisième phase est faite en simulation puis sur une mission de patrouille réelle.

La figure 5.1 rappelle la phase en ligne de l'approche proposée ainsi que les différentes étapes qui vont être détaillées dans la suite de ce chapitre. Elle consiste à exécuter la mission en appliquant initialement la SAR_0 calculée lors de la deuxième phase (estimation hors ligne de la performance). Une fois la mission lancée, une estimation en ligne de la faisabilité de la mission vis à vis des différents axes de performances se fait comme suit :

- **Axe booléen** : la mission est réalisable tant que les ressources exigées sont disponibles ainsi que leurs paramètres sont fixés dans les limites calculées lors des phases hors ligne (limite vitesse par zone, fréquences capteurs, période de calcul, etc).
- **Axe continu** : la mission est réalisable tant que les marges de performance des axes continus sont estimées positives à la fin de la mission (2.1,2.2). La marge estimée

$\widehat{Perf}_{SR, axe}^{ind_i}$ est mise à jour tout au long de la mission en fonction de la mesure des performances courantes et de la progression spatio-temporelle du robot.

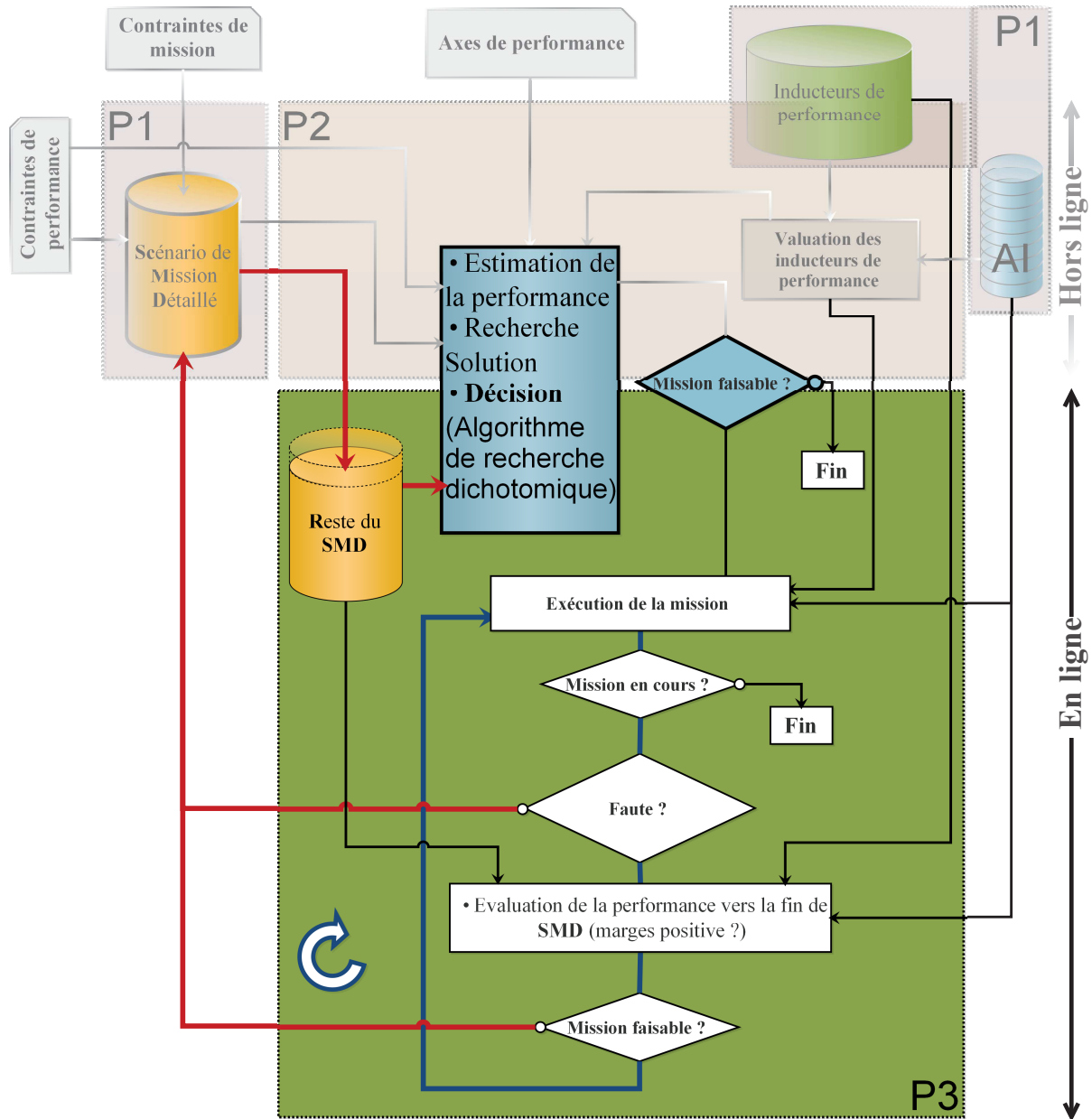


FIGURE 5.1 – Rappel de la phase en ligne (P3) de l'approche de gestion des ressources guidée par la performance

Si une faute ou une dérive de performance menant à une (des) marge(s) négative(s) est (sont) détectée(s), la mission est remise en cause et une nouvelle *SAR* doit être calculée pour poursuivre la mission. Un Reste de Scénario de Mission Détaillé *RSMD* est établi et utilisé par l'algorithme de recherche dichotomique afin de définir la *SAR* à appliquer (si elle existe).

Nous allons présenter maintenant l'outil de simulation utilisé avant de détailler les points significatifs du déroulement de la mission simulée.

5.1 Simulation

5.1.1 Description de l'outil de simulation

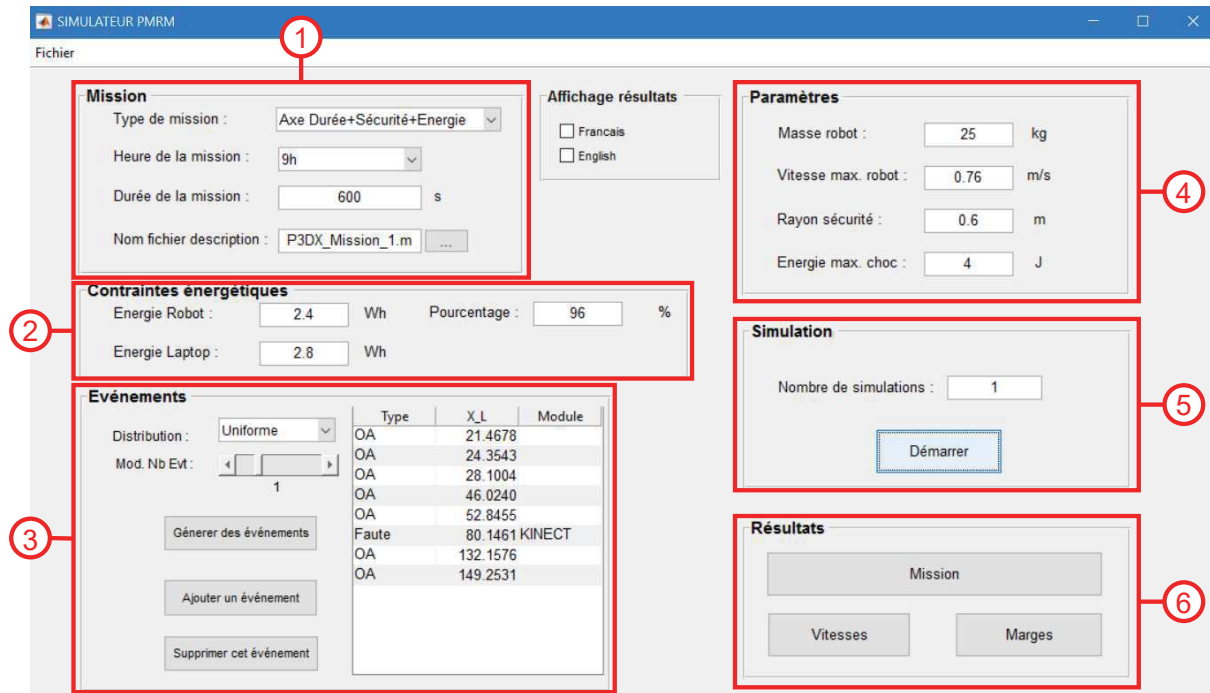
La validation de la méthodologie de gestion de ressources proposée est faite en premier lieu en simulation. Un simulateur a été développé sous *Matlab*. Il permet de simuler les différentes phases de la méthodologie proposée. Il a permis de faire une première validation de la méthodologie ainsi que des différents algorithmes correspondants aux trois phases de l'approche (projections de contraintes, décision, suivi de marges, etc.).

Dans le but d'ajouter un certain degré de réalisme sur la mission simulée un ensemble d'événements peut être généré aléatoirement. Ils peuvent correspondre à un ensemble de fautes sur des ressources matérielles et/ou logicielles ainsi que des évitements d'obstacles (EO) dont le nombre et les positions sont calculés par des lois probabilistes. Elles sont issues d'une analyse statistique du badgeage dans le laboratoire permettant de déterminer le nombre de personnes que le robot peut potentiellement rencontrer en fonction du lieu considéré et de l'heure choisie pour l'expérimentation [Ber15].

Les entrées du simulateur sont saisies via une interface constituée de 6 panneaux distincts (Figure 5.2) :

1. Le panneau **Mission** permet de décrire la mission à effectuer en renseignant les axes de performance pris en considération (Axe durée, axes durée/sécurité ou axes durée/sécurité/énergie), l'heure à laquelle celle-ci se déroule, la contrainte de durée $Perf_{C_{Dree}}$ ainsi que le nom du fichier de description de la mission (séquence d'objectifs du point de vue utilisateur) et des contraintes environnementales (présence humaine, couloir, vitres, etc.).
2. Le panneau **Contraintes énergétiques** permet de renseigner les contraintes énergétiques sur les batterie du PC $Perf_{C_{Energie}}^{PC}$ et du robot $Perf_{C_{Energie}}^R$. Celles-ci étant nécessaires lors d'un type de mission Axe durée/sécurité/énergie.
3. Le panneau **Événements** permet de générer des événements de façon aléatoire suivant une distribution choisie (Uniforme, Gaussienne ou Exponentielle). Il est aussi possible de générer des événements de façon manuelle à l'aide du bouton "Ajouter un événement". Le type d'événement (EO ou faute sur une ressource particulière) et la coordonnée linéaire de l'occurrence de l'événement sont également choisis.

4. Le panneau **Paramètres** permet de définir les paramètres structurels du robot tels que sa masse et sa vitesse maximale. On y définit aussi le rayon de sécurité (distance minimale par rapport aux obstacles) r_{SMZ} .
5. Le panneau **Simulation** permet de définir le nombre de simulations que l'on souhaite lancer et de les démarrer.
6. Le panneau **Résultats** permet de sélectionner les résultats que l'on souhaite afficher (Marges de performance, déroulement de la mission et évolution de la vitesse).


 FIGURE 5.2 – Interface du simulateur *Matlab* de mission avec gestion de la performance

Maintenant que l'outil de simulation de la mission et de l'approche de gestion de la performance est présenté, nous allons détailler par la suite le déroulement de la mission de patrouille simulée ainsi que les différents points significatifs.

5.1.2 Simulation

Les événements produits selon leurs distributions spatio-temporelle et leurs types, peuvent impliquer des déroulements assez variés pour une unique mission envisagée. Ainsi, parmi l'infinité de scénarios envisageables du déroulement de la mission simulée, nous en avons choisi un assez représentatif pour démontrer l'efficacité de l'approche proposée. Nous allons présenter dans cette section les événements générés pour la mission étudiée et détailler dans la suite les principaux points clés au cours de cette mission.

5.1.2.1 Génération d'événements

Un ensemble d'événements de type évitement d'obstacles (EO) ou fautes sur certains capteurs a été généré afin d'apporter des perturbations durant le déroulement de la mission et ainsi remettre en cause la solution initialement calculée dans le chapitre précédent. Le tableau 5.1 montre la liste d'événements, leurs coordonnées linéaires et les activités correspondantes. Conformément à la description des contraintes de la mission visée (section 3.2.1) les évitements d'obstacle ne peuvent se faire qu'en zones de présence humaine (salle d'expérimentation et couloir CR_1). Un évitement d'obstacle fait parcourir au robot une distance plus grande que celle prévue. Cela accroît la durée de la mission d'une durée supplémentaire qui est fonction de la vitesse v_r courante du robot et du rayon de contournement supposé égal à 1 m (Figure 5.3). Les fautes capteurs quant-à-elles ($laser_2$ puis US) se produisent sans contraintes spatio-temporelle.

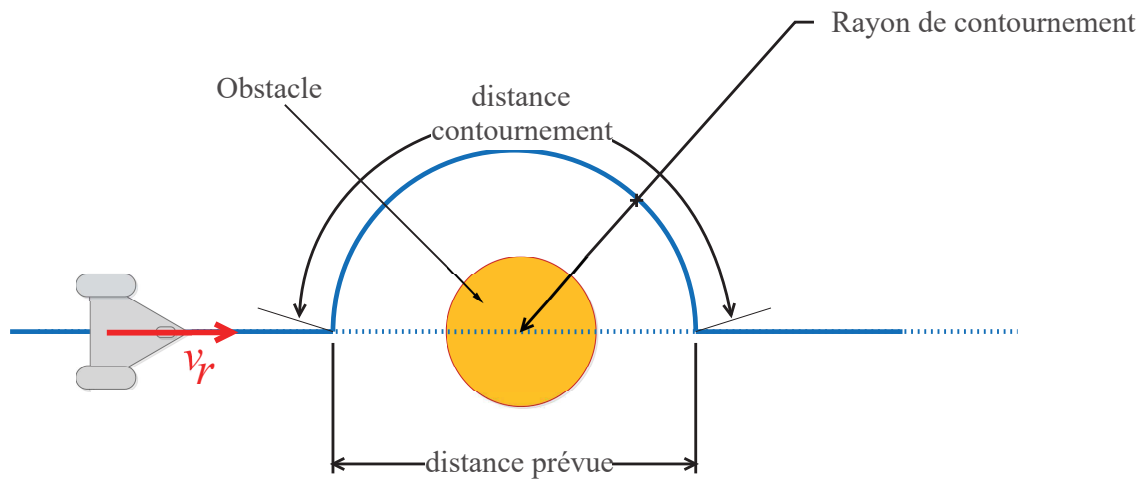


FIGURE 5.3 – Illustration de la prise en compte d'un évitement d'obstacle EO dans le simulateur

5.1.2.2 Déroulement de la mission simulée

Tous les axes de performance sont désormais considérés en dehors d'une analyse rigoureuse de la stabilité. La figure 5.4 montre les événements générés (EO et fautes) le long de la progression linéaire de la mission. La mission est initialement estimée réalisable avec la solution d'allocation de ressources SAR_0 calculée lors de la deuxième phase de l'approche (Table 4.17). Il faut noter que si la mission se réalise avec SAR_0 et qu'aucun événement ne se produit, les marges de performance estimées restent constantes tout au long de la mission.

Tableau 5.1 – Événements générés dans la mission simulée

Type d'événement	Coordonnée linéaire x_l	Activité correspondante
EO	3.15	A_1
EO	5.34	
EO	10.04	
EO	20.37	
EO	22.67	
EO	24,05	
EO	28,00	
EO	44.04	A_8
EO	46.35	
EO	49.13	
EO	51,91	
EO	55.36	
EO	60.95	
Faute <i>kinect</i>	118.11	A_{14}
EO	126.45	A_{15}
Faute <i>laser2</i>	142.02	
EO	150.65	A_{16}

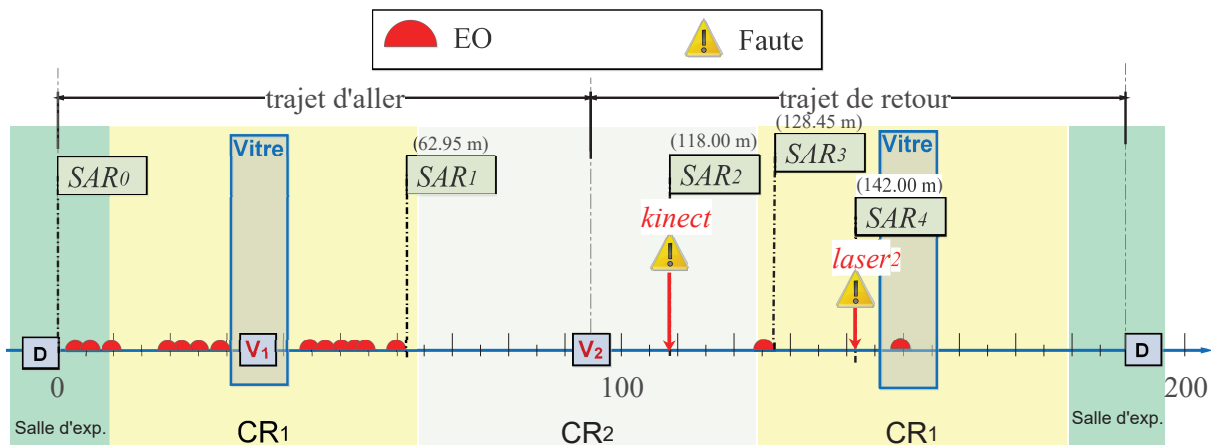


FIGURE 5.4 – Déroulement de la mission simulée

La mission est entamée avec le processus d'estimation et de gestion en ligne des performances (P3) (voir section 2.2.4). En total, 15 EO se produisent durant la mission. Initialement, le robot part du point D (salle d'expérimentation) à la vitesse maximale de l'activité initiale (0.4 m/s) (Figure 5.6). Les premières EO générés provoquent une baisse

des marges de durée, batterie robot et batterie du PC (Figure 5.5). Les marges de performances sont alors mises à jour en fonction des performances mesurées lors de l'exécution de la mission. La mission reste réalisable avec SAR_0 tant que les marges sont positives. Le passage dans la zone de vitre ($[100 \text{ s}; 150 \text{ s}]$ correspondant à $[31 \text{ m}; 41 \text{ m}]$) impose au robot de baisser sa vitesse à $v_{US_{max}}$. A_4 , A_5 et A_6 sont des activités sans progression linéaire ($x_l = 37 \text{ m}$ et $v_r = 0 \text{ m/s}$ (Figure 5.6(b))) mais à durées fixes (Figure 5.6(a)).

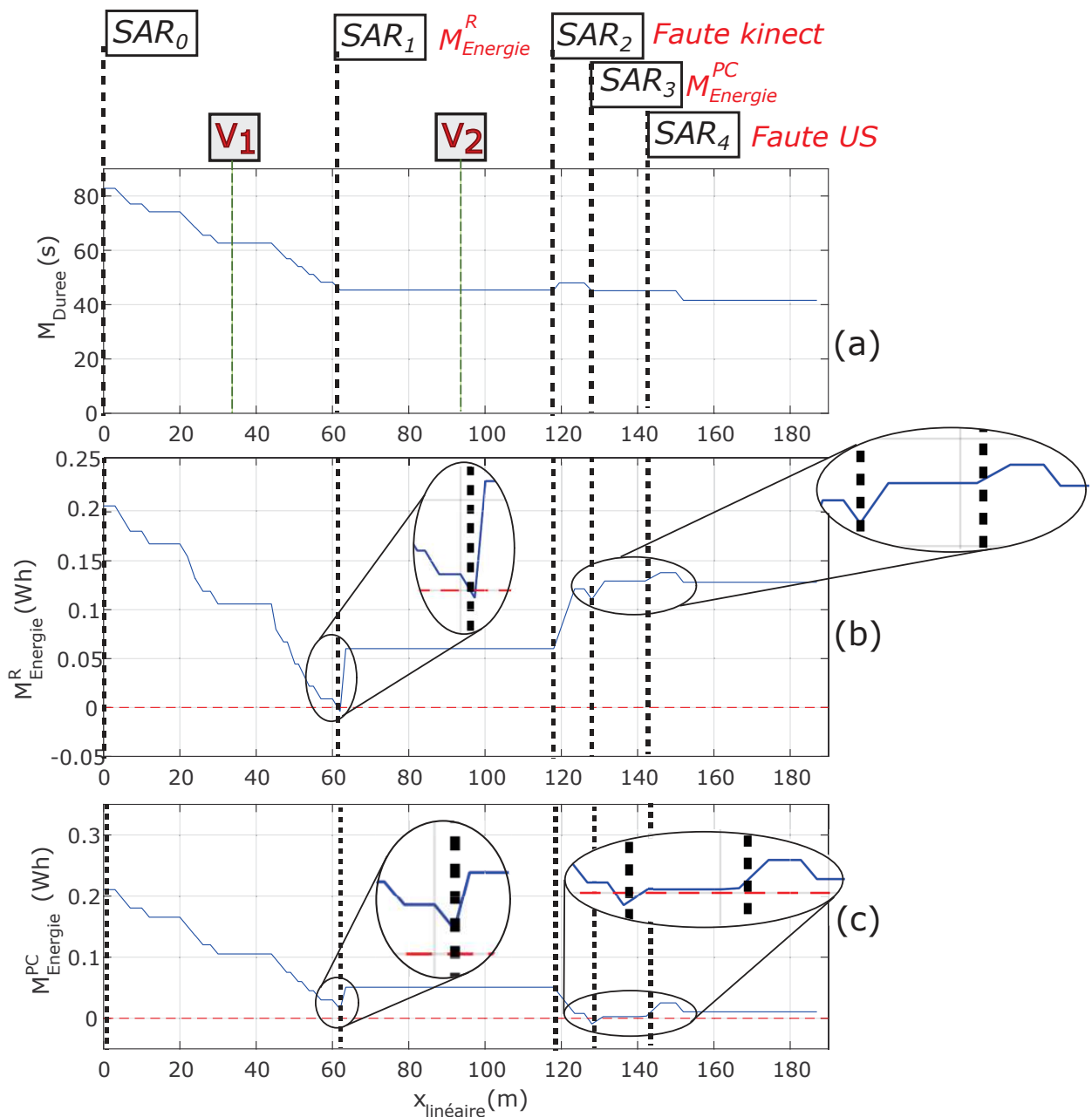


FIGURE 5.5 – Évolution des marges de performance continues de la mission simulée

Après 13 EO entre les activités A_1 et A_8 , la marge d'énergie sur la batterie du robot

$M_{Energie}^R$ devient négative (coordonnée linéaire $x_l = 62$ m dans la figure 5.5(b)). Donc si la mission se poursuivait avec les AI correspondantes à SAR_0 , la mission ne serait pas réalisable car le robot n'aurait pas assez d'énergie (batterie robot) pour l'achever même s'il n'y aura aucun événement. Une nouvelle SAR est à calculer en fonction de l'état courant du robot et de ses performances (l'état actuel de ses réserves d'énergie batterie robot et PC et de durée).

Le Reste de Scénario de Mission Détaillé $RSMD_1$ est ainsi établi (voir Annexe B). Comme aucune AI n'a été disqualifiée, le nombre d'AI éligibles reste inchangé pour les différentes activités restantes (de A_8 à A_{17}) par rapport au SMD calculé initialement (hors ligne). Le vecteur d'information de la première activité du $RSMD_1$ doit être actualisé afin de prendre en compte la progression du robot dans l'activité A_8 . La coordonnée de départ x_{A_8} ainsi que la distance dis_{A_8} sont alors recalculées.

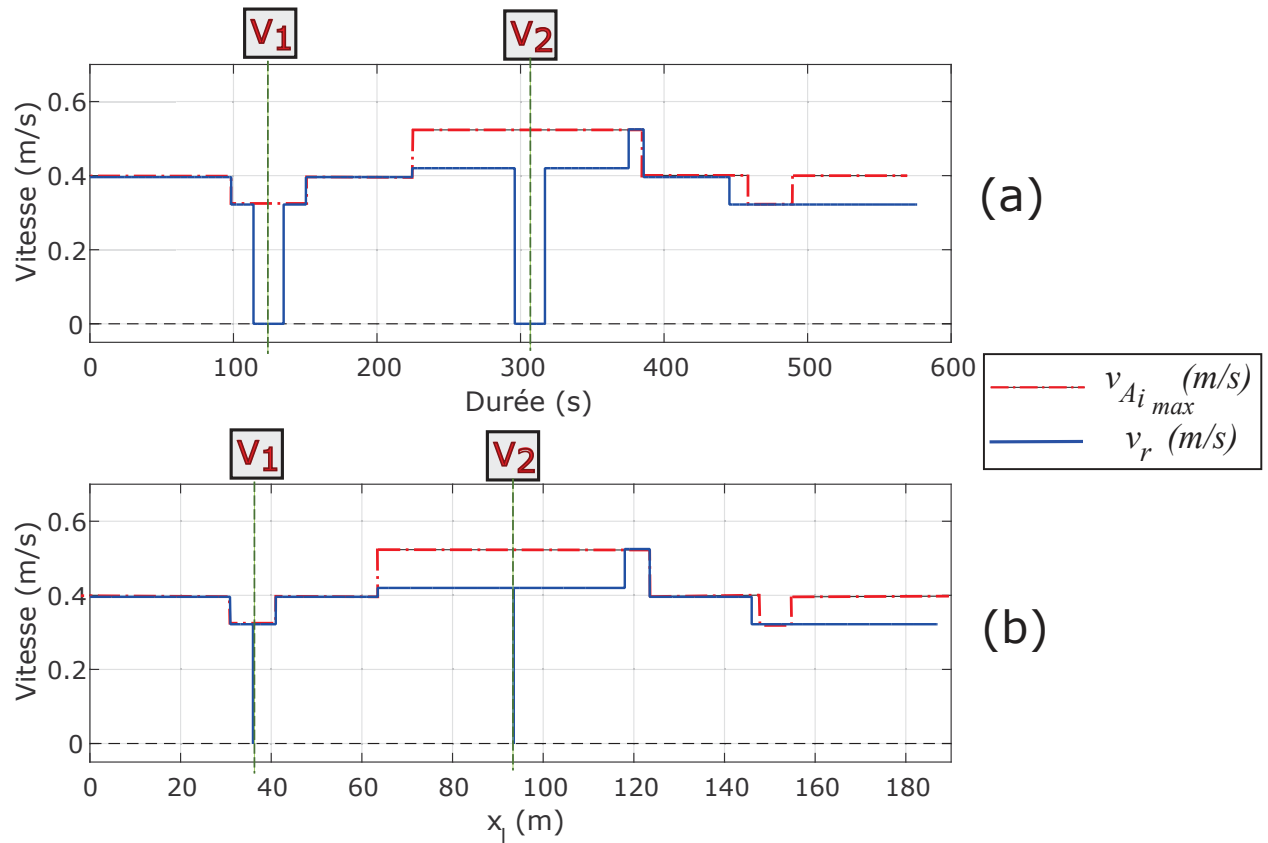


FIGURE 5.6 – Évolution de la vitesse du robot en fonction du temps et de la coordonnée linéaire x_l de position du robot

La SAR_1 détaillée dans le tableau 5.2 est ainsi calculée en se basant sur la description du reste de la mission $RSMD_1$. La complexité du problème est égale à $3 \cdot 10^9$ (calculée avec (2.8)). Le nombre d'itérations nécessaires pour trouver SAR_0 est $IT = 306$). Il est normal qu'elle soit inférieure à celle calculée pour SAR_0 ($3.3 \cdot 10^{14}$) puisqu'il ne reste que 10 activités à exécuter ($< n_{act} = 17$ pour SAR_0).

La solution d'allocation calculée remplace certaines AI par des AI moins énergivores. L'AI (11 : suivi de chemin avec évitement d'obstacles (SMZ) en utilisant les deux lasers et les sonars et une localisation basée sur les QR-codes en utilisant la kinect) est remplacé par l'AI (8 : SMZ avec un seul laser et les sonars avec localisation QR-codes) pour A_{16} . L'activité A_{17} va être exécutée selon SAR_1 avec l'AI (2 : SMZ avec uniquement les sonars et la localisation odométrique) au lieu de (1 : SMZ en utilisant un laser et les sonars ainsi que l'odométrie pour assurer la localisation) calculée par SAR_0 (voir les tableaux 3.1 et 4.9).

La marge d'énergie sur la batterie du robot $M_{Energie}^R$ (Figure 5.5(b)) redevient alors positive et la mission est à nouveau réalisable. Il faut remarquer que $M_{Energie}^{PC}$ augmente aussi. Ceci provient du fait que l'énergie consommée sur la batterie du PC par les AI (8) et (2) est inférieure à celle des AI (11 et 1). D'autre part, la marge de durée reste insensible à ce changement (jusqu'au prochain EO), vu que la vitesse du robot reste la même pour les deux AI.

Tableau 5.2 – SAR_1 calculée lors de la mission simulée

Activité A_i	A_1	$\cdots$	A_7	A_8	A_9	A_{10}	A_{11}		
SAR_1				11	11	11	23		
$dur_{A_{jmin}}$ (s)				24.21	66.66	4.76	10		
$E_{A_j^R}$ (Wh)				0.11	0.30	0.02	0.04		
$E_{A_j^{PC}}$ (Wh)				0.12	0.32	0.02	0.034		
Activité A_i	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	Perf. estimée		
SAR_1	25	23	11	11	8	2	-	-	
$dur_{A_{jmin}}$ (s)	1	10	71.42	56.82	31.05	96.87	555	54	
$E_{A_j^R}$ (Wh)	0.02	0.04	0.32	0.25	0.11	0.20	2.44	0.06	
$E_{A_j^{PC}}$ (Wh)	0.01	0.03	0.35	0.29	0.14	0.36	2.75	0.05	

La mission se poursuit avec SAR_1 et le robot entre dans la zone de non présence humaine pour atteindre la vanne V_2 . La vitesse du robot est limitée dans cette zone à v_{rmax} . Puisque la localisation est assurée par LQR (localisation basée QR-codes), et conformément à la contrainte c_5 , le robot ne doit pas dépasser $v_{QR-codes}$ (voir section 3.2.2). Puisqu'aucun EO ne peut se produire dans cette zone, toutes les marges restent

constantes. La vitesse linéaire du robot v_r s'annule pour les activités A_{11} , A_{12} et A_{13} (rotation vers V_2 et inspection de la vanne).

Le retour au point de départ D commence avec A_{14} en zone sans présence humaine (CR_2). À $x_l = 118$ m, une panne de *kinect* est simulée. Puisque ce capteur est utilisé pour l'activité courante A_{14} , A_{15} (AI (11)) et A_{16} (AI (8)), la SAR_1 n'est plus valide. Par conséquent, une nouvelle SAR doit être calculée avec prise en compte des différentes ressources restants éligibles. Le $RSMD_2$ est alors établie (voir Annexe B) et la liste des AI éligibles pour les activités restantes ne compte plus d'AI mobilisant la kinect. Le nombre de solutions possibles est de 21952 et la solution SAR_2 est trouvée après 108 itérations.

La SAR_2 est montrée dans le tableau 5.3. L'AI (11) prévue pour être utilisée dans A_{14} et A_{15} (SAR_1) est remplacée par l'AI (18) mobilisant les deux lasers et les sonars pour l'évitement d'obstacles (SMZ) et la méthode de Chromo-localisation LCH basée sur les données lasers au lieu de la localisation basée QR-codes utilisant la kinect. La même méthode de localisation est utilisée pour A_{16} avec un seul laser et les sonars (AI (15)). Comme pour SAR_1 , la dernière activité est réalisée avec l'AI (2).

 Tableau 5.3 – SAR_2 calculée lors de la mission simulée

Activité A_i	A_1	...	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	Perf. estimée	Marges
SAR_2				18	18	15	2	-	-
$dur_{A_{jmin}}$ (s)				10.47	56.82	31.05	78.28	552	48
$E_{A_j^R}$ (Wh)				0.05	0.21	0.10	0.18	2.38	0.12
$E_{A_j^{PC}}$ (Wh)				0.07	0.34	0.14	0.30	2.79	0.01

La nouvelle SAR calculée impacte les différentes marges de performance de la figure 5.5. Avec l'utilisation de l'option d'implémentation de tâche (OIT) LCH dans la zone de non présence humaine, le robot peut rouler à sa vitesse maximale. Cette vitesse est supérieure à $v_{QR-codes}$ imposée précédemment par l'utilisation de la localisation basée QR-code LQR . Ceci a engendré une augmentation de la marge de durée M_{Duree} . De même, la marge d'énergie sur la batterie du robot a augmenté car la kinect n'est plus utilisée. Cependant, la charge de calcul embarqué des différentes AI allouées pour le reste de la mission est supérieure à celle de SAR_1 . La marge d'énergie consommée sur la batterie du PC $M_{Energie}^{PC}$ a ainsi baissé tout en restant légèrement supérieure à 0.

La SAR_2 est maintenant appliquée. Mais comme $M_{Energie}^{PC}$ est très proche de 0, le premier EO généré dès le retour dans la zone de présence humaine CR_1 (A_{15}) conduit à son passage en négatif à $x_l = 128$ m. Suite à cet événement, le $RSMD_3$ est établi pour le reste de la mission (voir Annexe B). Le nombre de solutions possibles est de 1568. La SAR_3 (Table 5.4) est trouvée en 63 itérations. Elle permet de poursuivre la mission

en substituant l'AI (15) en (1). Le déplacement est toujours réalisé avec un laser et les sonars (avec SMZ). Cependant la localisation est faite uniquement par l'odométrie et par conséquent le deuxième laser ne va pas être utilisé. Ceci a permis de rendre la mission de nouveau réalisable vis-à-vis des différents indicateurs de performance.

Tableau 5.4 – SAR_3 calculée lors de la mission simulée

Activité A_i	A_1	$\cdots$	A_{14}	A_{15}	A_{16}	A_{17}	Perf. estimée	Marges
SAR_3				18	1	2	-	-
$dur_{A_{jmin}}$ (s)				45.45	31.05	78.28	555	45
$E_{A_j^R}$ (Wh)				0.08	0.08	0.18	2.67	0.13
$E_{A_j^{PC}}$ (Wh)				0.29	0.12	0.30	2.75	0.05

Néanmoins une faute sur le $laser_2$ est générée à $x_l = 142$ m. Comme ce capteur est utilisé dans l'activité courante (AI (18) pour A_{15}), la poursuite de la mission est à nouveau remise en cause. Le $RSM D_4$ est alors établi pour les 3 activités restantes. La liste des AI éligibles est filtrée afin de disqualifier les AI avec le $laser_2$ utilisé (voir Annexe B). Le nombre de solutions possibles est réduit à 18. Une nouvelle SAR_4 est calculée (Table 5.5). L'AI (1) est adoptée pour poursuivre l'activité courante A_{15} . Elle consiste en un suivi de chemin, avec évitement d'obstacle de type SMZ avec un seul laser et les sonars ainsi que l'odométrie pour assurer la localisation du robot.

La non utilisation du $laser_2$ pour le reste de la mission a apporté une augmentation de $M_{Energie}^R$ (Figure 5.5). De même, $M_{Energie}^{PC}$ augmente suite à ce changement, vu que l'AI (1) est aussi moins énergivore en coût calculatoire. Évidemment, la marge de durée reste constante tant que les vitesses adoptées pour les différentes activités reste inchangées.

Tableau 5.5 – SAR_4 calculée lors de la mission simulée

Activité A_i	A_1	$\cdots$	A_{14}	A_{15}	A_{16}	A_{17}	Perf. estimée	Marges
SAR_4				1	1	2	-	-
$dur_{A_{jmin}}$ (s)				10.10	31.05	78.28	555	45
$E_{A_j^R}$ (Wh)				0.04	0.08	0.18	2.36	0.14
$E_{A_j^{PC}}$ (Wh)				0.05	0.12	0.30	2.77	0.33

Il faut remarquer que la vitesse du robot a été limitée à $v_{US_{max}}$ même après avoir quitté la zone de vitre (A_{16}) et entamé la dernière activité A_{17} ($v_{A_{i_{max}}}$) (Figure 5.6). Ceci vient de l'utilisation de l'AI (2) qui fait appel à un suivi de chemin SMZ avec uniquement les sonars comme capteurs de proximité.

La mission simulée s’achève alors avec succès vu que les différentes marges de performance restent positives ($M_{Duree} = 41$ s, $M_{Energie}^{PC} = 0.12$ Wh, $M_{Energie}^{PC} = 0.01$ Wh) (axes continus). Quand aux axes de performance booléens, ils ont été respectés en utilisant les ressources indispensables avec les paramètres (vitesse) exigés. 5 SAR ont été calculées pour adapter le robot aux différents événements rencontrés tout au long de l’exécution de la mission (dérive de performance suite au EO et fautes des capteurs). Le tableau 5.6 récapitule les différentes SAR calculées avec en gras les AI différentes par rapport à la SAR précédente. Au total, 7 AI ont été sélectionnées parmi les 25 AI disponibles. L’AI courante a été changée au cours des activités A_{14} et A_{15} suite à l’occurrence de fautes sur des capteurs. Autrement, les changements d’AI ne concernent que les activités futures afin d’anticiper une situation problématique.

Tableau 5.6 – Tableau récapitulatif des différentes SAR calculées pour la mission en simulation

Activité A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
SAR_0	11	11	11	23	25	23	11	11	11	11	23
SAR_1								11	11	11	23
SAR_2											
SAR_3											
SAR_4											

Activité A_i	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
SAR_0	25	23	11	11	11	1
SAR_1	25	23	11	11	8	2
SAR_2			18	18	15	2
SAR_3				18	1	2
SAR_4				1	1	2

L’approche de gestion des ressources matérielles et logicielles a été validée en simulation. La phase d’estimation et gestion en ligne a permis de prévoir d’une façon anticipée une impossibilité de réalisation de la mission avant l’allocation courante des AI. Un $RSMD$ a été établi pour les différents cas et une SAR a été calculée en fonction de l’état de progression de la mission, des AI éligibles et des performances du robot. De plus, l’algorithme de recherche dichotomique a permis de trouver des SAR avec un nombre d’itérations relativement faible par rapport au nombre colossal de solutions possibles pour certains cas. Cette efficacité est d’une importance cruciale pour une implémentation de cet algorithme dans une architecture temps réel (section suivante). Enfin, il faut remarquer que dans l’exemple traité seules des fautes matérielles ont été simulées. Cependant, des fautes logicielles auraient été traitées de la même façon.

Cette validation a pris en compte différents types événements. Pourtant, certaines hypothèses sont irréalistes comme le suivi parfait du chemin (asservissement et erreur de localisation) à une vitesse constante ainsi que les estimations exactes des consommations et des durées des activités (erreur de modélisation). Pour démontrer que l'approche proposée reste tout aussi performante face à de véritables conditions expérimentales, nous allons aborder dans la suite la mission de patrouille réelle. Elle nous permettra de valider l'efficacité de l'approche proposée vis à vis les différentes perturbations et incertitudes dans une réalité de terrain.

5.2 Expérimentation

5.2.1 Conditions expérimentales

La validation expérimentale de l'approche de gestion des ressources du robot est abordée dans cette section pour la mission décrite dans la section 3.2. La réalité du terrain va amener différents types de perturbations tels que des évitements d'obstacles fixes ou dynamiques, des erreurs de localisation, des chemins avec des virages, des variations des vitesses durant la mission impliquant des accélérations/décélérations (changement de zones), etc. De façon à mettre en évidence le traitement d'une panne nous considérerons qu'à un instant donné une faute matérielle rend l'un des capteurs du robot inopérant.

La simulation faite précédemment a montré que le fait d'adopter une stratégie de maximisation de la consommation d'énergie sur la batterie du robot (sous une contrainte prédéfinie) implique souvent un passage de la marge correspondante en négatif suite au moindre évitement d'obstacle. Ainsi, pratiquement, une marge minimale de 4% a été imposée pour éviter de fréquentes remises en cause de la mission.

Nous allons ensuite détailler les points significatifs du déroulement de la mission ainsi que les différentes remises en cause de la faisabilité de celle ci.

5.2.2 Déroulement de la mission

La SAR_0 est initialement appliquée ([vidéo de la mission](#)¹). Le robot entame la première activité en effectuant des virages lui permettant de quitter la salle d'expérimentation et d'atteindre le couloir CR_1 (Figure 3.18).

1. <https://www.youtube.com/watch?v=jdTiLjeKC78>

Contrairement à la simulation, la saturation des actionneurs lors des virages ainsi que les phases d'accélération ont conduit le robot à rouler à des vitesses souvent inférieures à la consigne (Figure 5.7). La figure 5.8 montre les relevés expérimentaux des puissances consommées sur les batteries du robot et du PC ainsi que l'évolution des énergies délivrées par celle-ci. Ces puissances théoriquement considérées comme des constantes ne le sont pas expérimentalement. Cela est dû aux oscillations de la trajectoire du robot pour suivre son chemin comme le montre la figure 5.9 et par conséquent les accélérations nécessaires pour le maintenir sur le chemin.

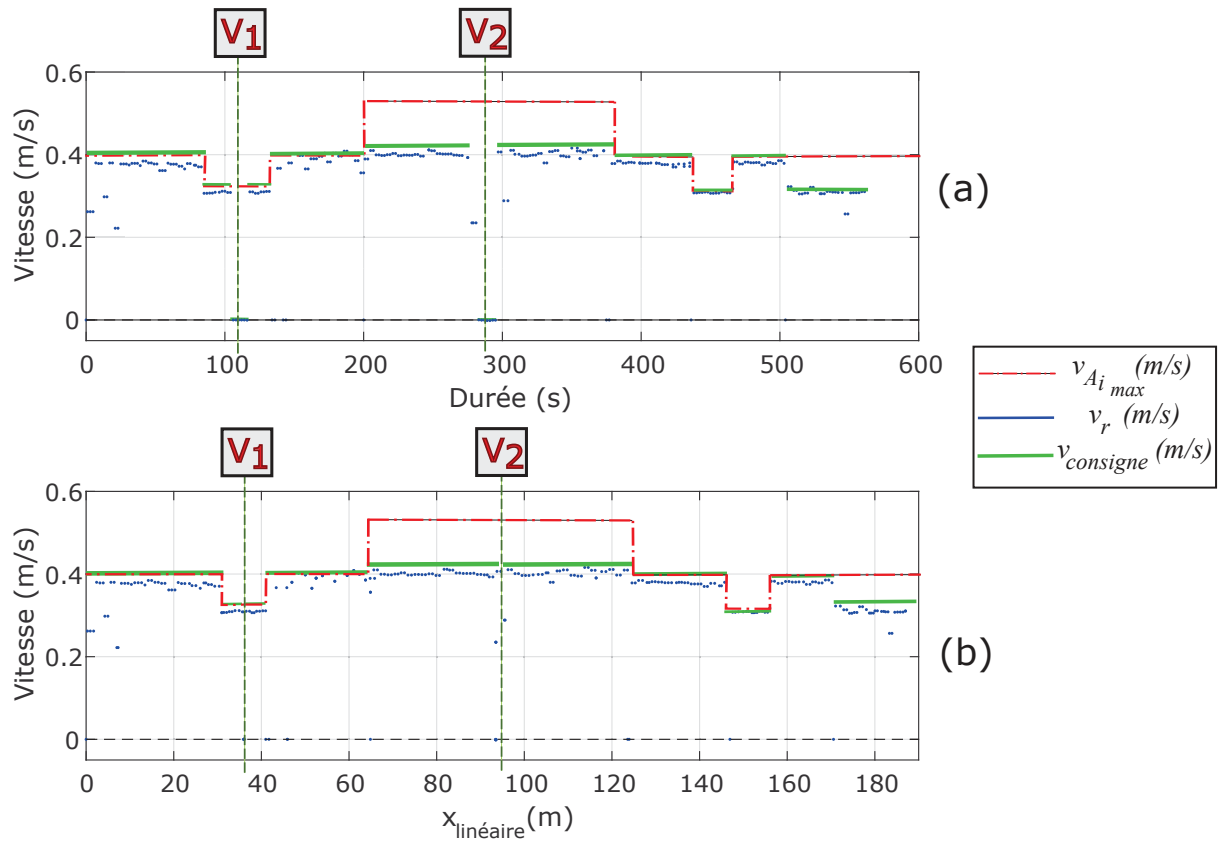


FIGURE 5.7 – Evolution de la vitesse linéaire du robot au cours de la mission de patrouille

Ceci explique la décroissance de $M_{Energie}^R$ observées dans la figure 5.10 et, dans une moindre mesure celle de $M_{Energie}^{PC}$ et M_{Duree} . Les différentes marges continuent à décroître jusqu'à ce que le robot effectue les activités de rotation et d'inspection de la vanne V_1 . Les activités A_4 et A_6 sont exécutées pratiquement avec une durée inférieure à celle supposée a priori constante car le cap du robot à l'arrivée vers $x_l = 36$ m est d'ores et déjà quasiment orienté vers V_1 . Ce gain en durée et en énergie explique les hausses des différentes marges. Le robot réduit sa vitesse à $v_{US_{max}}$ dans la zone vitrée comme le montre la figure 5.7.

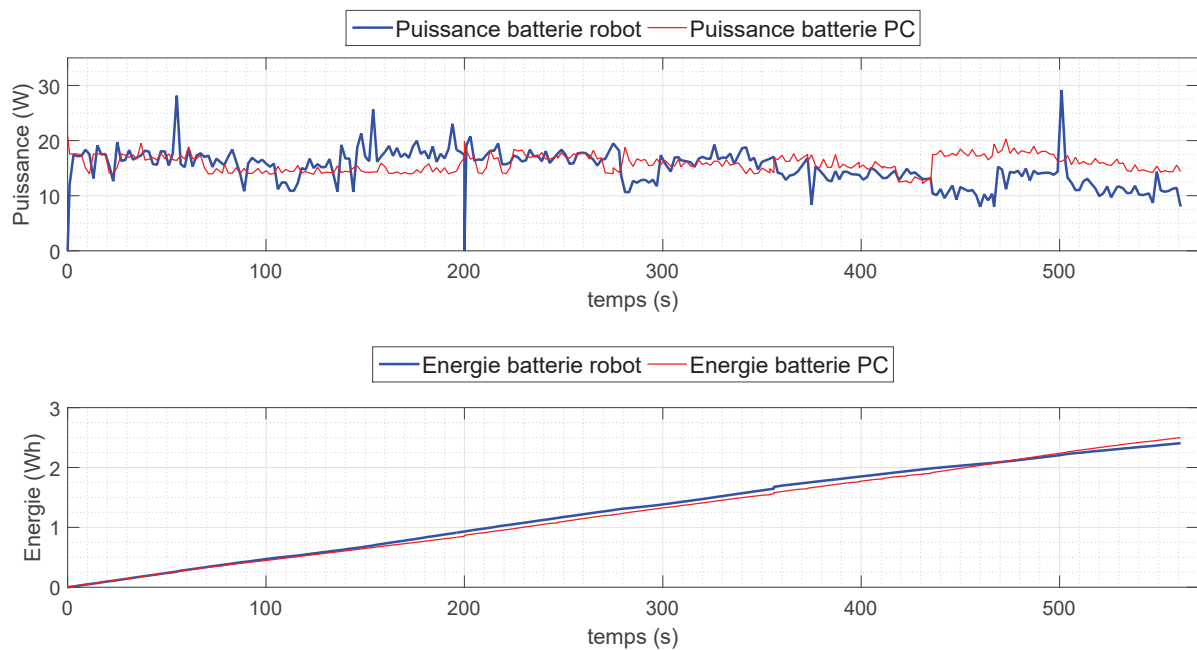


FIGURE 5.8 – Puissance et énergie consommées sur les batteries du robot et du PC

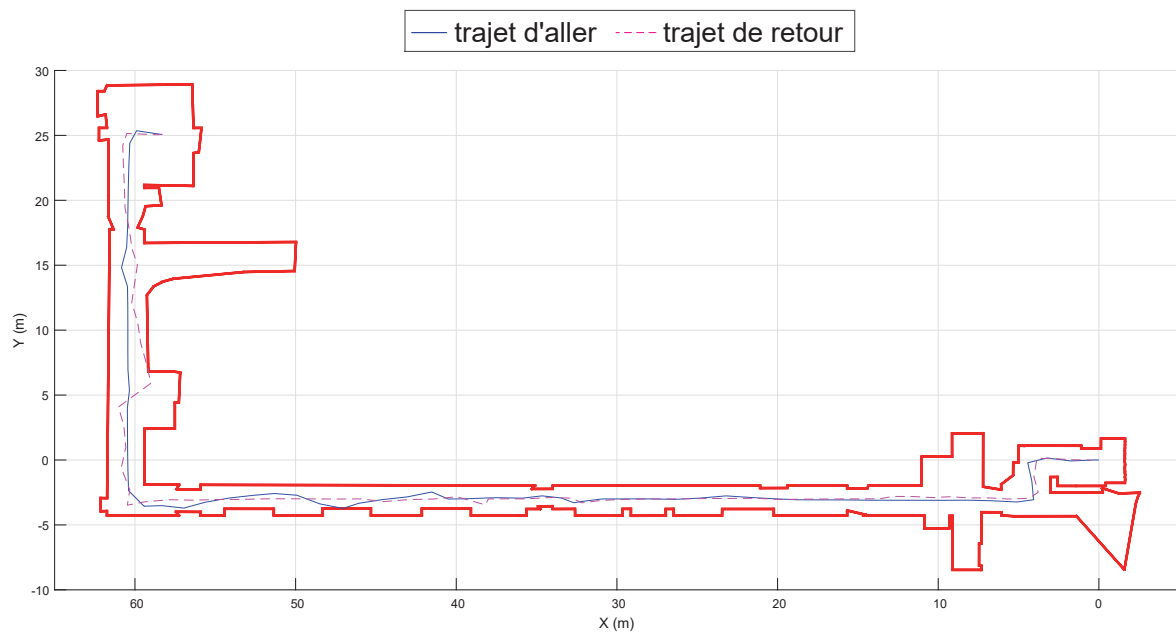


FIGURE 5.9 – Trajectoire du robot lors de la mission de patrouille : elle est construite à partir des données odométriques recalées lors du passage à proximité des QR-codes

La mission se poursuit après l'inspection de V_1 . $M_{Energie}^R$ continue de décroître jusqu'à devenir négative à $x_l = 63$ m (Figure 5.10). La poursuite de la mission avec SAR_0 est donc estimée impossible. Dans ce cas le $RSMD_1$ est établi et l'algorithme de recherche de solutions d'affectation est à nouveau utilisé pour trouver la SAR_1 qui permet de poursuivre la mission (nombre d'itérations 479). Le tableau 5.7 montre les AI allouées pour exécuter les deux dernières activités (en gris). $M_{Energie}^R$ redevient positive et la mission reste réalisable.

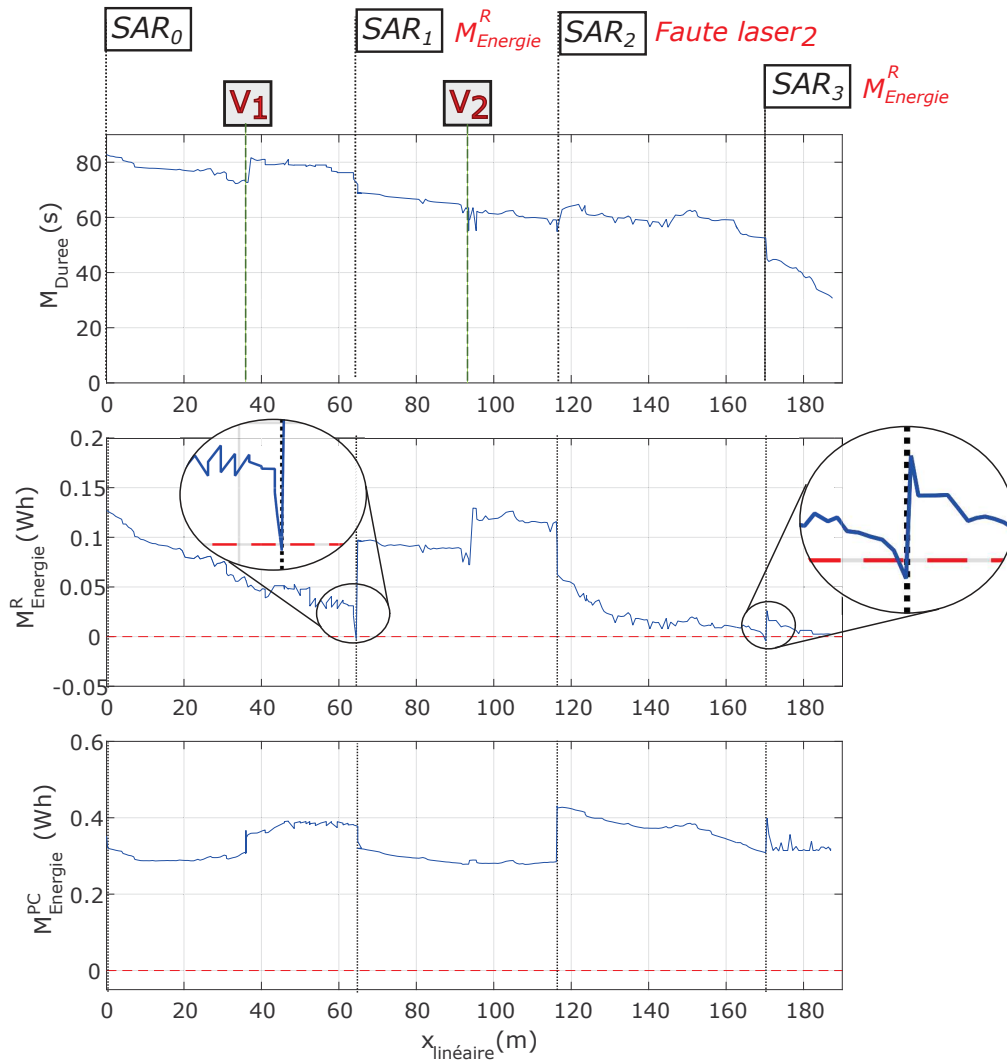


FIGURE 5.10 – Évolution des marges de performances continues lors de la mission de patrouille

Le robot atteint le deuxième couloir CR_2 et puisque la localisation est assurée par l'OIT LQR sa vitesse est limitée à $v_{QR-codes}$ dans la zone sans présence humaine. Le robot atteint la vanne V_2 , et pour les mêmes raisons que pour V_1 , une augmentation de

$M_{Energie}^R$ est remarquée. Le retour vers le point de départ, après l'inspection de V_2 , est alors entamé. Toujours dans CR_2 le robot roule avec la vitesse linéaire consigne $v_{QR-codes}$. Aucun obstacle dynamique n'est rencontré dans cette zone et pourtant les différentes marges décroissent. Comme il a déjà été expliqué, ceci est dû au fait que la vitesse consigne n'est pratiquement pas atteinte. De plus, une erreur de cap du robot et/ou la dérive odométrique, l'entraîne rapidement vers un évitement (d'obstacle) d'un des murs des couloirs provoquant des variations d'accélérations importantes.

Tableau 5.7 – SAR_1

Activité A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
SAR_1								11	11	11	23
$dur_{A_{jmin}}$ (s)								24.21	66.66	4.76	10
$E_{A_j^R}$ (Wh)								0.11	0.30	0.02	0.04
$E_{A_j^{PC}}$ (Wh)								0.12	0.32	0.02	0.034
Activité A_i	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	Perf. estimée		Marges		
SAR_1	25	23	11	11	9	2					
$dur_{A_{jmin}}$ (s)	1	10	71.42	56.82	31.05	96.87					
$E_{A_j^R}$ (Wh)	0.02	0.04	0.32	0.25	0.13	0.20					
$E_{A_j^{PC}}$ (Wh)	0.01	0.03	0.35	0.28	0.15	0.36					

Vers la fin de A_{14} à $x_l = 116$ m (256 s), une panne est simulée sur le $laser_2$. Ce capteur étant utilisé dans l'AI (11) et (1), SAR_1 n'est plus valable et par conséquent la mission est à nouveau remise en cause. Comme pour les précédents cas, un $RSMD_2$ est établi (voir Annexe B). Une SAR_2 est trouvée (103 itérations) et la mission peut être poursuivie (Table 5.8). Les 3 AI correspondants aux A_{14} , A_{15} et A_{17} sont différentes par rapport à ceux de SAR_1 . Pour A_{17} , la non utilisation du $laser_2$ a permis d'activer le $laser_1$ et d'utiliser la localisation basée QR-codes au lieu de celle purement odométrique. Cette solution moins coûteuse en énergie sur la batterie du PC, est plus énergivore sur la batterie du robot. Ceci explique les variations des différentes marges suite au calcul de SAR_2 (Figure 5.10).

Tableau 5.8 – SAR_2

Activité A_i	A_1	...	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	Perf. estimée	Marges
SAR_2				8	8	9	8	-	-
$dur_{A_{jmin}}$ (s)				18.93	56.82	31.05	72.28	537	63
$E_{A_j^R}$ (Wh)				0.07	0.21	0.13	0.30	2.44	0.06
$E_{A_j^{PC}}$ (Wh)				0.08	0.26	0.15	0.36	2.37	0.43

Le robot exécute les AI déterminées par SAR_2 durant les activités A_{14} , A_{15} et A_{16} et le début de A_{17} . Cependant, la marge d'énergie sur la batterie du robot devient de nouveau négative à $x_l = 171$ m. La mission se poursuit avec SAR_3 qui correspond, comme le montre le tableau 5.9, à l'AI de la dernière activité. Le $laser_1$ est ainsi désactivé. Les sonars sont donc les uniques capteurs proximétriques utilisés. La vitesse est réduite à $v_{US_{max}}$ (M_{Duree} diminue). $M_{Energie}^R$ devient positive et $M_{Energie}^{PC}$ augmente vu que l'AI (9) est moins énergivore que l'AI prévue d'être utilisée par SAR_2 .

 Tableau 5.9 – SAR_3

Activité A_i	A_1	$\dots$	A_{16}	A_{17}	Perf. estimée	Marges
SAR_3				9		
$dur_{A_{j_{min}}} (s)$				73.80	557	43
$E_{A_i}^R (Wh)$				0.21	2.47	0.03
$E_{A_i}^{PC} (Wh)$				0.33	2.4	0.4

À l'arrivée du robot les différentes marges sont positives ($M_{Duree} = 37$ s, $M_{Energie}^{PC} = 0.31$ Wh, $M_{Energie}^{PC} = 0.01$ Wh). Les contraintes imposées avant la mission sont respectées. 3 SAR ont été calculées pour adapter les configurations du robot courantes ou futures en raison des marges estimées négatives et d'une panne de capteur. Le tableau 5.10 montre les différentes SAR calculées ainsi que les AI exécutées (en bleu). L'AI courante a été reconfigurée à deux reprises (A_{14} et A_{17}).

 Tableau 5.10 – Tableau récapitulatif des différentes SAR calculées au cours de l'expérimentation

Activité A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
SAR_0	11	11	11	23	25	23	11	11	11	11	23
SAR_1								11	11	11	23
SAR_2											
SAR_3											

Activité A_i	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
SAR_0	25	23	11	11	11	1
SAR_1	25	23	11	11	9	2
SAR_2			8	8	9	8
SAR_3						9

L'expérimentation abordée dans cette section a permis de valider l'approche de gestion des ressources matérielles et logicielles du robot. Grâce à ce mécanisme d'estimation de reconfiguration dynamique et en ligne, la mission est devenue à nouveau réalisable à plusieurs reprises après avoir été estimée non réalisable. Cependant, cette adaptation de la *SAR* ne peut être toujours possible vis à vis des différents axes de performance. Ceci dépend évidemment de la position spatio-temporelle des événements et de leur nature. Par exemple, si un nombre important d'évitement d'obstacles se produit au début de la mission conduisant à une marge négative, une *SAR* est susceptible d'être trouvée en substituant des AI correspondant à des activités futures. Cependant, si le même nombre d'évitement d'obstacles a lieu vers la fin de la mission et qu'aucune alternative n'est possible à l'AI courante, la mission ne pourra être poursuivie.

Un autre exemple peut être évoqué au sujet des pannes sur les ressources. Si un capteur/actionneur ou un algorithme est indispensable dans une zone (les sonars dans la zone de vitre) ou est unique pour réaliser une activité (la caméra kinect pour l'inspection de la vanne) ou encore est fautif avant l'atteinte de la zone ou activité considérée alors, la mission ne pourra se finir. La redondance matérielle et/ou logicielle est donc une dimension importante à prendre en compte si l'on ne veut pas faire face à une telle situation. Autrement, et tel que nous avons montré dans les exemples précédents, le même événement peut être surmonté tant qu'un minimum de redondance subsiste [Jai+16b].

5.3 Conclusion

Ce chapitre a détaillé la phase dynamique de l'approche de gestion des ressources proposée. Nous avons tout d'abord détaillé sa mise en œuvre sur la mission de patrouille en simulation. Différentes perturbations ont été générées aléatoirement afin de mettre en évidence les différents mécanismes d'estimation et de reconfiguration des allocations courantes et futures des ressources matérielles et logicielles du robot.

Malgré la remise en cause de la réalisation de la mission vis à vis des différents axes de performance et à plusieurs reprises l'algorithme de gestion des ressources a permis de trouver une *SAR* permettant la réalisation de la mission en fonction de l'état courant des performances, de la progression du robot et d'un scénario établi pour le reste de la mission. L'algorithme de recherche dichotomique a permis de trouver en temps réel une *SAR*, à plusieurs reprises, parmi un nombre colossal de solutions possibles.

Les expérimentations ont montré qu'en présence d'une ressource fautive, le recouvrement est fait objectivement en respectant les contraintes induites par les différents axes de performance. Le choix d'une solution alternative redondante n'est donc pas réalisé heu-

ristiquement comme le fait souvent l'opérateur humain en s'appuyant sur son expérience du domaine. De plus, ce choix dépendra des circonstances de la mission.

Outre les fautes matérielles et/ou logicielles, ce chapitre a démontré la robustesse de l'approche proposée envers d'autres types de "fautes" telles que les erreurs de modélisations (estimation des performances) et le manque de connaissances sur l'environnement (obstacles statiques inconnus ou dynamiques). Le concept de marges de performance a apporté un degré d'immunité envers ces différents types de fautes [Jai+16b].

Conclusion générale

Les travaux présentés dans ce manuscrit s'intéressent aux Systèmes Cyber-Physiques dans le domaine de la robotique mobile. Ces derniers se caractérisent par une autonomie accrue et doivent mener des missions complexes et de longues durées au sein d'un environnement dynamique. Plus précisément notre travail s'est focalisé sur une problématique nécessairement centrale dans ce cadre d'étude, mais pourtant rarement abordée : l'identification des ressources matérielles et logicielles devant être mobilisées, par un système robotique autonome, tout au long de sa mission, pour garantir que cette dernière vérifie un ensemble de contraintes de performance, même en présence de défaillances matérielles et/ou logicielles, de perturbations externes (évitement d'obstacle) ou d'aléas internes (erreurs de mesures, etc.).

Notre analyse nous a tout d'abord amené à nous interroger sur la notion d'autonomie qui nécessairement est très fortement liée à celle de choix. Celui-ci nécessite de disposer d'un critère objectif permettant de l'assurer. C'est tout naturellement que nous nous sommes tournés vers le concept de performance, qui, bien que couramment utilisé en robotique, ne disposait pas d'un cadre clairement accepté et défini en robotique mobile. Nous avons alors proposé une décomposition de la performance multicritère en axes principaux (Sécurité, Energie, Localisation, Stabilité) indispensables à la réalisation de toute mission en robotique mobile, et en axes secondaires relevant des attentes de l'utilisateur (Durée).

Le cœur de notre travail a été la proposition d'une démarche holistique d'allocation dynamique des ressources matérielles et logicielles d'un système robotique effectuant une mission contrainte en terme de performance. Cette démarche se décompose en 3 phases. La première, hors ligne, identifie tout d'abord, pour chaque axe de performance, les indicateurs et inducteurs de performance ainsi que les modèles d'estimation associés. Elle construit ensuite le Scénario de Mission Détaillée (SMD), qui est un élément central de notre proposition. Ce dernier décompose le scénario initial, dans l'espace et le temps, en une succession d'activités où les objectifs visés sont réalisés sous un ensemble de contraintes (environnementales, matérielles, règlementaires, de performance) constantes. Chacune des activités identifiées peut alors être mise en œuvre par un ensemble d'Alternatives d'Implémentation (AI) au sein duquel il faut choisir et paramétrer les ressources

matérielles et/ou logicielles qui doivent être utilisées lors de l'exécution de la mission.

Vient alors la phase, toujours hors ligne, où il convient de conclure quant aux solutions d'affectation de ressources à choisir et paramétrer. Nous avons démontré dans ce manuscrit que le problème adressé pouvait se formaliser sous la forme d'un problème de sac-à-dos multicritère et donc de complexité NP complet. Pour faire face à l'explosion combinatoire rencontrée, et de façon à pouvoir trouver une bonne solution d'affectation en un temps compatible avec les contraintes temps réel imposée en robotique mobile, nous avons alors adapté à notre problématique l'algorithme dichotomique proposé par Bennour [Ben+05] sur l'affectation des ressources humaines aux processus d'entreprise. Ce dernier nous permet alors de conclure très rapidement sur la capacité (ou non) du système robotique utilisé à exécuter la mission sous les contraintes de performance imposées. Si cela est possible une Solution d'Allocation de Ressources (SAR) est alors identifiée. Nous sommes donc en mesure de garantir à l'utilisateur que le robot peut (ou non) exécuter sa mission en respectant les contraintes de performance imposées. Ce travail se distingue notablement des travaux de Nyons sur la garantie de performance qui ne permettent que d'estimer la probabilité de succès d'une mission sans proposer d'adaptation aux déviations ou défaillances rencontrées. Enfin, la dernière phase consiste à suivre en ligne, tout au long de l'exécution de la mission, l'évolution réelle de la performance par rapport à celle prévue. En cas de dérive de performance trop importante induite par les approximations de nos modèles, des événements imprévus (évitement d'obstacle), ou des dysfonctionnements matériels ou logiciels, une nouvelle SAR est déterminée en temps réel, si nécessaire et si elle existe, pour le Reste du Scénario de Mission Détaillé (RSMD).

La robustesse de notre approche repose sur un concept très important pratiquement, la définition de Marges de Performance qui constituent autant de réserves de performance (temps, énergie) qui nous permettent d'absorber les inévitables dérives de performance et aléas de déplacement (évitement d'obstacle non planifié).

Par ailleurs, une des forces de notre démarche est de proposer une méthode de reconfiguration objective des ressources allouées en présence de dérive énergétique majeure, ou de défaillance matérielle et/ou logicielle imprévue. Elle participe donc à l'amélioration de la tolérance aux fautes du système robotique. Dès la détection de tels aléas, une solution de recouvrement est déterminée en temps réel, même si les schémas impactés n'allaient peut-être être utilisés que bien plus tardivement lors de l'exécution du scénario de la mission. Dans certains cas nous pouvons donc anticiper l'impact d'une ressource défaillante avant qu'il ne soit observé.

Enfin, une forte originalité de notre étude est d'avoir adressé de façon détaillée l'autonomie énergétique qui, bien qu'étant incontournable, n'est que très rarement abordée en robotique mobile. En s'appuyant sur les travaux de Parasuraman [Par+14] nous avons

construit un modèle générique permettant d'exprimer la puissance consommée tant au niveau de la batterie du robot, que de celle du PC embarqué supportant l'architecture de contrôle. Ce modèle intègre l'impact des capteurs, actionneurs et algorithmes utilisés. Son évaluation démontre qu'il affiche une précision acceptable (10%) compatible avec l'expérimentation.

L'ensemble de la démarche a été déployée et validée avec succès sur une mission de patrouille d'environ 200 m d'un robot Pioneer P3-DX, dans les couloirs du laboratoire, en environnement dynamique. L'approche proposée a été détaillée pour les axes de performance sécurité, énergie et durée. Le développement d'un simulateur, en facilitant l'exploration d'un large éventail d'événements perturbateurs (évitement d'obstacle, pannes matérielles et/ou logicielles), a permis de valider avec succès notre proposition. Pour terminer l'expérimentation a confirmé l'efficacité de l'approche d'allocation des ressources matérielles et logicielles proposée.

Compte tenu de l'ensemble des points que nous venons de résumer, il semble clair que ce travail contribue à satisfaire les objectifs d'auto-optimisation (pilotage de la performance), auto-configuration (adaptation aux circonstances) et auto-guérison (recouvrement) que nous avons énoncés, en introduction, pour les Systèmes Cyber-Physiques. Par conséquent il cherche à conférer au système robotique, dans une certaine mesure évidemment, des propriétés de sécurité (axe sécuritaire de performance), de durabilité (axe énergétique de performance, performance assurée), de résilience (maintient de la performance même en présence d'aléas), d'efficacité (stratégie d'allocation des ressources matérielles et logicielles), de prédictibilité (mission scénarisée), de robustesse, fiabilité et sûreté de performance (gestion des défaillances et aléas de la mission).

Mais, en l'état, bien évidemment, ce travail souffre d'un ensemble de limitations que nous n'avons pu aborder, faute de temps, mais qu'il conviendrait d'adresser.

La dimension énergétique est un axe d'étude pour lequel nous sommes confrontés à deux problèmes épineux :

Le premier, et non des moindres, est la lourdeur de l'identification des modèles énergétiques du robot et du laptop. Cela nécessite de très nombreux relevés expérimentaux pour toutes les configurations matérielles et/ou logicielles pouvant être utilisées. D'une part chaque nouvel ajout matériel ou logiciel nécessite une nouvelle campagne d'identification. D'autre part, certains de ces relevés peuvent devenir caducs pour la batterie robot si l'on change notablement sa masse, où pour la batterie laptop, si l'on modifie de façon conséquente l'environnement logiciel d'exécution de l'architecture.

Le second n'est pas lié directement à notre méthodologie mais plutôt aux limites actuelles des connaissances qui ne permettent pas de connaître avec précision la quantité

exacte d'énergie contenue (State of Charge (SoC)) dans une batterie. Il est clair que cette information est primordiale dans le cas de missions autonomes. En effet la batterie étant un dispositif électrochimique complexe son SoC dépend d'un nombre très important de paramètres (composition chimique, nombre de cycles de charge et décharge, température, état de santé, condition d'usage, etc.) [RP03]. Dans notre cas, pour déterminer la capacité énergétique de nos batteries, nous avons mesuré la quantité d'énergie fournie après une charge complète, jusqu'à leur décharge totale. Puis nous avons supposé cette valeur constante.

Mais l'axe énergétique est aussi une dimension pour laquelle nous pourrions assez facilement approfondir et étendre le travail déjà réalisé.

Tout d'abord, dans nos travaux, nous avons considéré que le robot se déplaçait sur un sol plan et de coefficient de frottement uniforme. En s'appuyant sur les travaux de Sadrpour *et al.* [SJU12; SJU13b; SJU13a] qui intègrent dans leur modèle énergétique l'impact à la fois de la nature du sol et de son inclinaison, nous pensons qu'il serait possible d'étendre notre modèle énergétique à des environnements plus variés que celui dans lequel nous avons évolué. Cependant il resterait à démontrer expérimentalement que les modèles énergétiques de déplacement élaborés garderaient une précision acceptable compatible avec la capacité d'anticipation (qui nécessite une connaissance *a priori* de l'environnement) que nous devons préserver pour déployer notre approche d'allocation des ressources.

Ensuite, il est naturel de s'interroger sur l'hypothèse forte de vitesse constante de déplacement adoptée au niveau des activités d'une mission. Elle est utilisée dans de très nombreux travaux [SJU13b; SJU13a] car elle simplifie la formulation de la consommation énergétique de déplacement en éliminant les termes liés à l'accélération. Cependant expérimentalement elle n'est jamais rigoureusement vérifiée en raison des rampes présentes lors des changements de vitesse ou des variations de vitesse lors des changements de direction du robot pour asservir sa trajectoire ou éviter un obstacle. Par ailleurs, si nous nous plaçons purement dans une logique de minimisation énergétique, les travaux de Kim [KK08a] et Tokekar *et al.* dans [TKI14] ont démontrés analytiquement l'allure que devait avoir le profil de vitesse pour qu'un robot puisse se déplacer entre deux points à énergie minimale. Il faudrait donc s'assurer que théoriquement et expérimentalement le gain énergétique soit significatif par rapport à notre approche simplificatrice et cela en prenant en compte la contrainte de localisation qui n'est pas présente dans les travaux cités.

Une autre hypothèse forte de notre travail en lien avec la limitation précédente est la nature du chemin suivi pour réaliser la mission. Il est clair qu'il peut être plus ou moins long et que de ce fait il influence l'énergie qui sera consommée. Nous avons supposé que ce chemin est composé d'un ensemble de segments de droite. Si les travaux de Kim [KK08a]

ne se sont intéressés qu’au profil de vitesse à énergie minimale pour un segment de droite, Tokekar *et al.* dans [TKI14] ont étendu la résolution à un chemin quelconque. Comme précédemment, nous pourrions intégrer ces travaux dans notre approche. Nous pensons cependant que la décomposition d’un chemin en une succession de segments de droite permet d’adresser un large panel de missions surtout en environnement structuré.

D’autres limitations portent sur la dimension sécuritaire de notre approche.

La première, encore en lien avec la dimension énergétique, est le choix de la méthode d’évitement d’obstacle. En effet, l’approche SMZ retenue dans notre travail s’appuie sur un cercle de sécurité, qui, compte tenu des contraintes sécuritaires, se doit d’avoir un rayon assez important (0.8 m). En raison de l’environnement relativement confiné dans lequel nous avons évolué (largeur minimal du couloir de 1.8 m), et de l’erreur de localisation du robot engendrée par une dérive importante, il rentrait assez rapidement en comportement d’évitement d’obstacles. Bien qu’il puisse poursuivre sa mission, il parcourt ainsi une distance bien supérieure à celle prévue, avec des variations de vitesses importantes. Une façon de réduire ce type de comportement en environnement confiné est d’adopter un mécanisme d’évitement d’obstacle utilisant une zone sécuritaire faisant appel à une zone sécuritaire répartie de façon non homogène autour du robot de façon à mieux épouser la configuration de l’environnement. Par exemple du pourrions faire appel à la méthode de Zone Virtuelle Déformable (Deformable Virtual Zone) proposée dans Zapata *et al.* [ZCL04] qui permettrait d’utiliser une zone sécuritaire en forme, par exemple, d’ellipse dont le grand axe serait orienté selon le sens de déplacement du robot et le petit axe orthogonalement. Cela permettrait de traverser des zones confinées en minimisant l’engagement d’évitement d’obstacle.

Une limitation encore plus importante de notre travail sur le plan sécuritaire est que nous n’avons pu, faute de temps, établir le modèle dynamique de notre robot. En effet le modèle cinématique n’est valide que si le robot est léger et qu’il se déplace à faible vitesse avec des accélérations limitées. Si ces hypothèses ne sont pas vérifiées alors, en présence de brusques changements de vitesse, il y aura des erreurs de prédiction de la position du robot puisque ce dernier ne peut changer instantanément de vitesse. Ainsi expérimentalement lors d’engagement d’évitement d’obstacle nous avons dû faire face à des chocs inopinés avec l’environnement stoppant automatiquement le robot. Cela a bien évidemment impacté la robustesse de notre mission. Pour éviter de tels comportements il faut prendre en compte la dynamique du robot dans le mécanisme de navigation. Pour cela nous pourrions nous inspirer des méthodes proposées par Ivanjko *et al.* dans [IPP10] ou par Dhaouadi *et al.* dans [DH13]. Nous pensons que la prise en compte de la dynamique du robot risque de nous conduire à réduire encore la vitesse du robot, mais on ne peut transiger avec la sécurité. Cette intuition demande à être confirmée. Quoiqu’il en soit il faut absolument

engager ce travail d'expérimentation pour fiabiliser le mécanisme d'évitement d'obstacle et par la même l'exécution de notre mission.

Par ailleurs, on peut aussi se demander si l'on ne pourrait pas étendre le concept de marge de performance qui est utilisé pour les axes Energie et Durée à l'axe Sécuritaire. C'est une question que nous nous sommes posée. Bien que ce soit un axe booléen, il serait peut-être pertinent de quantifier un niveau de sécurité en prenant en compte la redondance matérielle et/ou logicielle permettant de maintenir une exécution sécuritaire de la mission même en présence de défaillance de ressources contribuant à assurer cette dimension de la performance.

Il est clair, aussi, que la scénarisation "déterministe" de notre mission induit aussi certaines limitations.

D'une part notre approche nécessite d'être en capacité de décrire l'enchaînement des objectifs devant être atteints durant l'exécution de la mission mais aussi de disposer d'un critère temporel global. En effet, c'est ce dernier qui nous permet de déduire les vitesses d'avance du robot.

D'autre part nous ne considérons actuellement que des missions "déterministes" dont l'évolution est planifiée du début à la fin et dont les objectifs finaux ne peuvent évoluer. Il serait pourtant très intéressant de pouvoir changer les objectifs en fonction des circonstances. Par exemple si l'on revient sur la mission que nous avons détaillée dans ce manuscrit il serait pertinent de détecter, en cours de mission, que celle-ci ne peut s'achever en l'état faute d'énergie par exemple en raison de très nombreux évitements d'obstacles, mais qu'il en reste cependant assez pour revenir à la base de recharge. Bien que la détection de l'occurrence de ce point de retour soit totalement asynchrone, cela ne remet absolument pas en cause l'item précédent. Il suffit de disposer d'une "base de missions" composée d'une mission de "patrouille" et d'une mission de "retour à la base" et de commuter de l'une à l'autre au moment opportun en paramétrant la seconde en conséquence. Voilà qui ouvre de nouveaux horizons en permettant d'aborder des scénarios bien plus complexes et globalement non déterministes.

Enfin, toujours en lien avec la notion de scénarios, il est aisé de constater qu'en multipliant les objectifs, les ressources et les contraintes de tous ordres, nous allons multiplier les activités ainsi que la combinatoire du problème d'allocation qui nous intéresse. Il est donc logique de s'interroger sur la combinatoire limite que nous serons capables de traiter en temps réel. En simulation, nous avons étendu notre mission en planifiant plusieurs allers et retours. Une solution d'allocation a été trouvée par l'algorithme dichotomique que nous utilisons en moins de 800 itérations au sein d'un espace de 10^{23} états. Nous sommes donc confiants dans notre capacité à pouvoir adresser expérimentalement des missions

bien plus longues et complexes que celle abordée dans ce manuscrit. Cependant si l'espace d'états s'agrandissait encore plusieurs solutions de replis pourraient être envisagées. En premier lieu il est important de rappeler que la première solution d'allocation est calculée hors ligne et que, par conséquent, pour celle-ci, la contrainte temps réel n'existe pas. On peut donc espérer que le prochain calcul d'affectation de ressources interviendra assez tardivement dans l'exécution de la mission pour que l'espace d'états restant à balayer soit compatible avec les contraintes temps réel. Dans le cas contraire, il nous faudrait envisager décomposer la mission initiale en sous-missions d'espace d'états de dimensions acceptable. Les contraintes sécuritaires et de durée peuvent être sans trop de difficultés rapportées au niveau de ces sous-missions. En revanche, la dimension énergétique pose plus de problème car il faudrait trouver un mécanisme permettant de répartir objectivement la contrainte énergétique globale au niveau de chacune des sous-missions.

Nous terminerons la partie consacrées aux limitations en abordant un certains nombre de points hétérogènes qui devraient être considérés.

Le premier se rapporte à l'amélioration de l'algorithme d'allocation utilisé pour résoudre le problème de sac-à-dos qui nous intéresse. Pour obtenir de meilleures solutions, c'est-à-dire des solutions préservant un maximum choix possibles pour chacune des activités Bennour [Ben+05] propose deux pistes. D'une part il préconise de faire appel à un mécanisme de retour en arrière qui, une fois une solution trouvée, revient sur les activités déjà considérées pour améliorer le résultat d'allocation. D'autre part il faut avoir en tête que l'ordre de prise en compte des activités est absolument arbitraire. Il n'est absolument pas certain que les traiter, comme nous le faisons, dans l'ordre chronologique inverse d'apparition dans la mission, conduise à la meilleure solution. Au niveau d'une activité des critères tels que le nombre de choix envisageables ou que l'écart maximal en performance devraient être pertinents pour guider l'ordre de traitement des activités dans l'algorithme d'allocation.

Le second concerne les fonctions d'utilité que constituent les lois qui permettent de choisir les valeurs pertinentes des inducteurs de performance. Celles-ci correspondront par transitivité à des valeurs pertinentes de la performance elle-même. Pour faire un choix il est indispensable de disposer d'un nombre fini d'alternatives. En présence de fonctions continues il en existe une infinité. Il faut donc pouvoir en extraire des points caractéristiques pertinents. Si cela peut relever parfois du bon sens, comme dans notre cas le choix de la vitesse minimisant l'énergie pour une longueur de déplacement donné, il faut aussi pouvoir choisir d'autres points balayant l'ensemble de l'espace envisageable. Il semble, en première approximation, souhaitable de disposer d'une valeur minimale, d'une valeur maximale, et d'une valeur de compromis qui n'est peut-être pas toujours aisée à déterminer. Enfin, pour terminer tournons nous vers l'automatique. Le choix, au

niveau de chacune des activités d'une mission, des ressources logicielles et matérielles à impliquer est pour l'instant effectué sans s'assurer que le changement de schémas de contrôle entre deux activités ne conduit pas à des situations critiques. En effet, il est nécessaire d'assurer la stabilité des schémas individuels, mais la phase de transition entre deux modes de fonctionnement doit aussi faire l'objet d'une étude de stabilité. Celle ci devra s'inspirer des résultats sur la stabilité des systèmes commutés qui fait actuellement l'objet de recherches intensives [LAS15].

Même s'il souffre encore des nombreuses limitations que nous venons de balayer, il reste encore un certain nombre de pistes d'étude.

Il paraît tout d'abord logique d'aborder en détail au plus vite l'axe principal de la performance qu'est la Localisation et dont l'étude n'a pu être conduite à son terme. Bien évidemment dans le travail présenté ici, cette dimension n'a pas été absente. Beaucoup d'informations ont enrichi la carte de la mission, en précisant par exemple les zones vitrées (qui imposaient l'utilisation des sonars), ou en spécifiant que l'algorithme de QR-Code devait être impérativement utilisé au voisinage des vannes pour se localiser avec précision. Cependant la contrainte de performance de Localisation n'a pas été explicitée et donc l'analyse de cette dimension n'a pu être menée à son terme. Nous pensons la formuler de la façon suivante : Quelles ressources utiliser tout au long de la mission de façon à ce que la position du robot soit connue avec une précision fixée par l'utilisateur, mais pouvant varier en fonction des zones traversées ? Pour cela il faudra quantifier précisément la précision de localisation que l'on peut atteindre en fonction des algorithmes utilisés, des capteurs impliqués, et de l'environnement traversé. Nous pensons pouvoir nous inspirer des travaux de Tunstels dans [Tun07] pour évaluer la performance de robots martiens. Évidemment la vitesse du robot et de la boucle de contrôle du robot, ainsi que sa dérive odométrique, la nature de l'environnement, et les capacités des capteurs seront autant de paramètres à prendre en compte. Il faut aussi remarquer l'impact que l'équipement de l'environnement, en ajoutant des QR-Codes par exemple, peut avoir une influence importante sur la précision de localisation qu'il sera possible d'atteindre. Bien évidemment le point excessivement dur qu'il restera à aborder est celui de la performance de Stabilité.

En effet, une modification de la période d'application du contrôle a, intuitivement, un effet considérable sur tous les axes de performance. L'axe énergétique est impacté. S'il est démontré, par exemple, qu'un doublement de la période d'échantillonnage du contrôle permet d'assurer la stabilité du système, la conséquence sera une réduction notable non seulement de la charge calculatrice du contrôle, mais aussi une diminution du taux de recrutement des capteurs, principale source de consommation énergétique. Les conséquences sur l'axe Localisation sont aussi importantes, en regard des arguments précédemment cités. L'axe sécurité est aussi impacté, du fait de la modification de la réactivité du système.

La question du choix de la période de contrôle du point de vue de la stabilité est un problème encore ouvert en particulier pour les systèmes non linéaires que sont les robots mobiles. Nul doute que les travaux récents sur les systèmes cyber-physiques permettront des avancées importantes sur ce thème.

Un autre point essentiel est à prendre en compte est la considération simultanée de plusieurs inducteurs dynamiques de performance ayant des domaines de variation continus. Dans le travail présenté nous ne jouons que sur la vitesse en supposant que la fréquence de recrutement des US a été fixée arbitrairement par ailleurs. Si plusieurs inducteurs de performance dynamiques doivent être considérés en même temps, il faudra définir une stratégie de valuation permettant de discrétiser le domaine de variation en préservant les solutions les plus pertinentes. Le recours à des approches de calcul par intervalles [Sun09] pourrait s'avérer utiles pour aborder cette problématique.

Enfin il est indispensable de relier le travail que nous avons réalisé à la mise en place d'une véritable architecture de contrôle tolérante aux fautes. En effet comme nous l'avons déjà énoncé, la démarche que nous avons proposée ne concerne que la phase de recouvrement du processus de tolérance. Il faut donc adjoindre en amont celle de détection et identification de la faute et intégrer le tout au sein de l'architecture. Il nous semble naturel de nous tourner vers le travail réalisé au LIRMM dans le cadre de la thèse de B. Durand [Dur11] [CGDL15] et qui d'ailleurs ne proposaient aucun critère de choix objectif pour la sélection des ressources matérielles et logiciels à utiliser. Limitations à laquelle nous avons donc répondu dans le cadre de cette thèse.

Pour terminer il nous semble indispensable d'évoquer les champs applicatifs du travail que nous venons de présenter.

Tout d'abord, nous pensons que le travail réalisé est maintenant assez mature pour aborder des missions bien plus longues et complexes. Les patrouilles en totale autonomie avec inspection de cibles et retour d'informations sont en effet une problématique industrielle totalement d'actualité. Celle présentée dans ce manuscrit nous a été inspirée par le projet Argos² lancé par le groupe pétrolier Total et auquel nous nous sommes intéressés. Dans le monde, un nombre très importants de sites d'extraction sont entièrement automatisés. Il faut donc développer des robots capables de les surveiller en totale autonomie, mais dans des conditions climatiques parfois extrêmes. Le modèle énergétique que nous avons établi, ainsi que notre méthodologie d'allocation des ressources matérielles et logicielles devraient nous permettre d'adresser cette problématique de surveillance avec succès. Nous pourrions aussi proposer et expérimenter des stratégies de couverture bien plus réalistes que celles proposées dans [Wei+12] par exemple et qui s'appuient sur un mo-

2. <http://www.argos-challenge.com/fr>

dèle énergétique très grossier. Il nous semble aussi que notre travail serait très intéressant dans le cadre de missions d'exploration. Évidemment la découverte d'un environnement qui constitue la première phase d'une mission d'exploration ne relève absolument pas de la problématique que nous avons adressée. En revanche, la plupart du temps, le robot doit être capable de revenir à sa base. Si les réserves énergétiques initiales sont connues et monitorées, il nous semble alors possible de déterminer à quel moment cesser l'exploration pour basculer vers une mission de retour à la base au sein de l'environnement qui vient d'être exploré pour laquelle notre travail peut être exploité. Évidemment nous ne disposons pas actuellement de planificateur de mission mais la problématique d'exploration nous semble être à portée de main.

Pour clore ce manuscrit il nous semble intéressant de s'interroger sur la capacité de notre approche à être portée vers d'autres classes de robots et à être adaptée à d'autres milieux.

Évidemment nos travaux relatifs à la dimension énergétique s'appuient sur un modèle de consommation des moteurs à courants continus. Fort heureusement ce type d'actionneurs est couramment utilisé dans nombre de robots qu'ils soient terrestres, drones aériens ou sous-marins. La maîtrise énergétique suppose d'être, dans un premier temps, en capacité de construire les modèles de consommation associés aux modes de déplacement du robot et, dans un second temps de suivre l'évolution de la consommation d'énergie en cours de mission. Cela implique d'embarquer sur le robot des wattmètres ou des capteurs de courant. Fort de notre expertise, nous espérons pouvoir construire des modèles énergétiques associés aux différents modes de déplacement du robot hexapode R.HEX³ de grande taille (60 kg) développé au sein de l'équipe Explore pour l'intervention en milieu naturel fragile. Toutes ses articulations sont d'ores et déjà munies de capteurs de courant. Il paraît par ailleurs indispensable d'engager la même démarche pour les robots sous-marins développés en collaboration avec notre équipe dans le cadre du projet étendard ALEYIN de l'université de Montpellier.

Il est très tentant de chercher à étendre notre méthodologie d'affectation des ressources matérielles à des robots autonomes aériens, sous-marins, voire d'exploration extra-terrestre. Même s'il est toujours souhaitable de confier plus d'autonomie aux systèmes artificiels de façon à alléger la charge du contrôle humain l'environnement aérien nous semble être celui pour lequel notre travail aura le plus de difficulté à se projeter. En effet même si sur le fond nous pensons que notre démarche peut y être adaptée il nous semble difficile de construire un modèle énergétique de déplacement crédible. Le milieu aérien nous semble être celui affecté de la plus grande hétérogénéité avec des flux d'air inconstants et aléatoires rendant excessivement difficile d'établir des lois de consommation

3. http://explore.lirmm.fr/?page_id=95

énergétique assez fiables pour envisager de bâtir des scénarios de missions assez robustes.

L’exploration extra-terrestre nous semble très proche du travail que nous venons de réaliser [AD10] [Sch+09b] [HAM09] [Tun07] [Ing+07]. Les conditions d’évolution sont voisines des conditions terrestres et présentent une certaine homogénéité, le besoin d’autonomie est avéré, l’optimisation de l’utilisation des ressources est indispensable, la tolérance aux fautes y est recherchée, la dimension énergétique est incontournable. Autant d’aspects que nous abordons dans le cadre de notre étude et qui pourraient intéresser l’ESA (Agence Spatiale Européenne).

Enfin, pour terminer, l’environnement sous-marin semble tout indiqué pour ouvrir nos travaux vers un autre élément. C’est en effet un milieu qui, sous bien des aspects, s’apparente au milieu spatial. Par ailleurs, même s’il existe des flux de liquide, les conditions d’évolutions restent bien plus homogènes que dans l’air. L’emploi d’AUV (Autonomous Underwater Vehicle) est difficile en robotique sous-marine car l’autonomie est très difficile à déployer dans ce milieu hostile [Set12] [KC13] [Woi14]. Contrairement au milieu aérien (ou spatial), en raison du faible débit d’informations envisageable dans l’eau (100 Kbits/s) il est impossible de téléopérer le robot sans avoir recours à un lien physique. On fait le plus souvent appel à des ROV (Remotely Operated underwater Vehicle) qui sont reliés à l’opérateur par un long câble de liaison. Ce lien physique procure de nombreux avantages (supervision en temps réel de la mission par l’opérateur, sauvegarde des données, lien énergétique). Malheureusement ce dernier, dans le cadre de certaines missions dont l’exploration en environnement confiné (e.g. karstique [Car05] [Gar+08] [GA10]), induit de nombreux problèmes (longueur limitée, perturbation du déplacement surtout lors du retour à la base) qui parfois sont rédhibitoires au regard des objectifs. Il est donc indispensable de pouvoir affranchir le robot de ce lien physique pour qu’il puisse évoluer en totale autonomie tout en garantissant que ses objectifs de performance énergétique, de sécurité, de localisation et de tolérance aux fautes soient bien garantis. Là encore, voilà autant de défis que le travail que nous venons de présenter dans cette thèse peut contribuer à satisfaire.

Annexe A

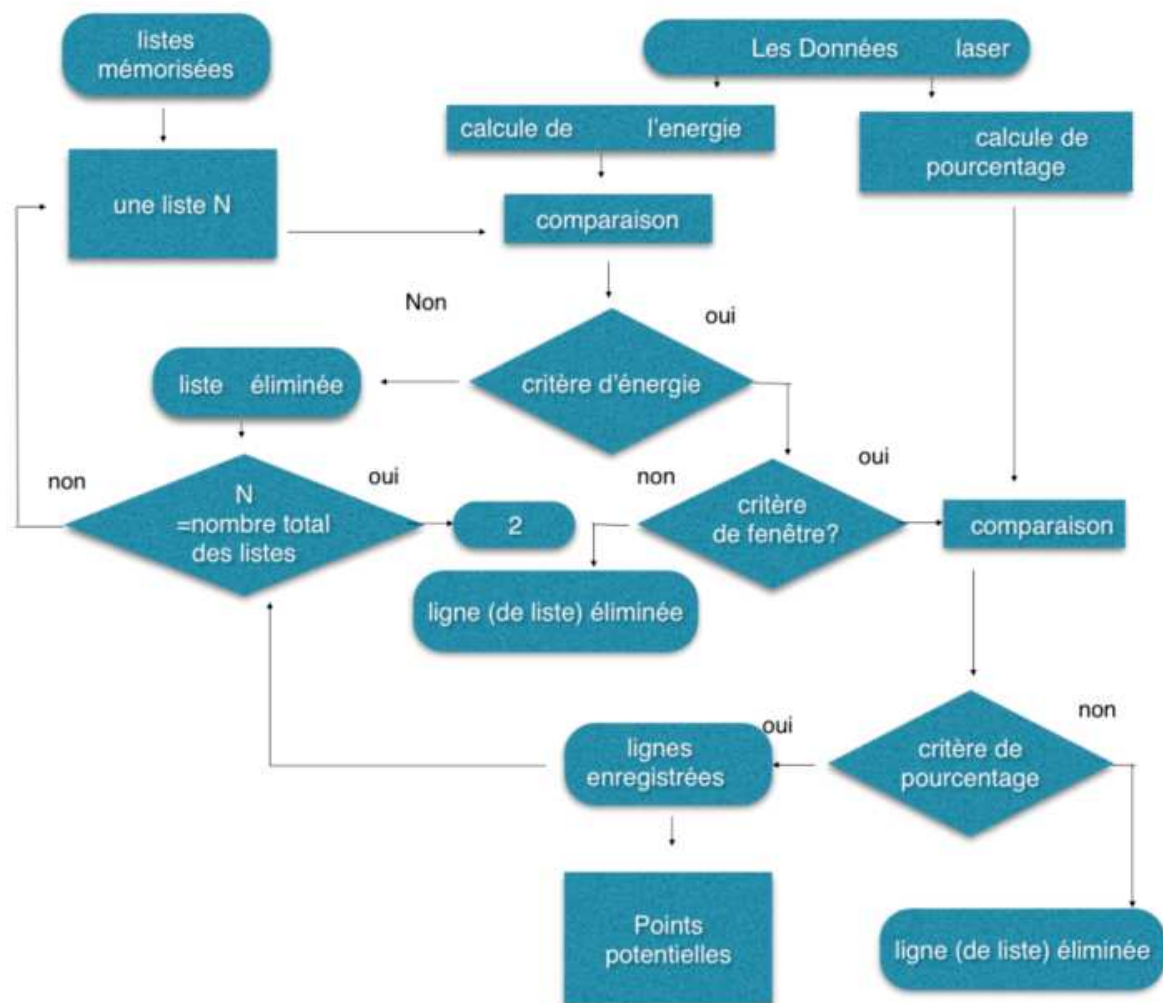


FIGURE 5.11 – Première partie de l’algorithme de l’algorithme LZA (figure tirée de [Mou14])

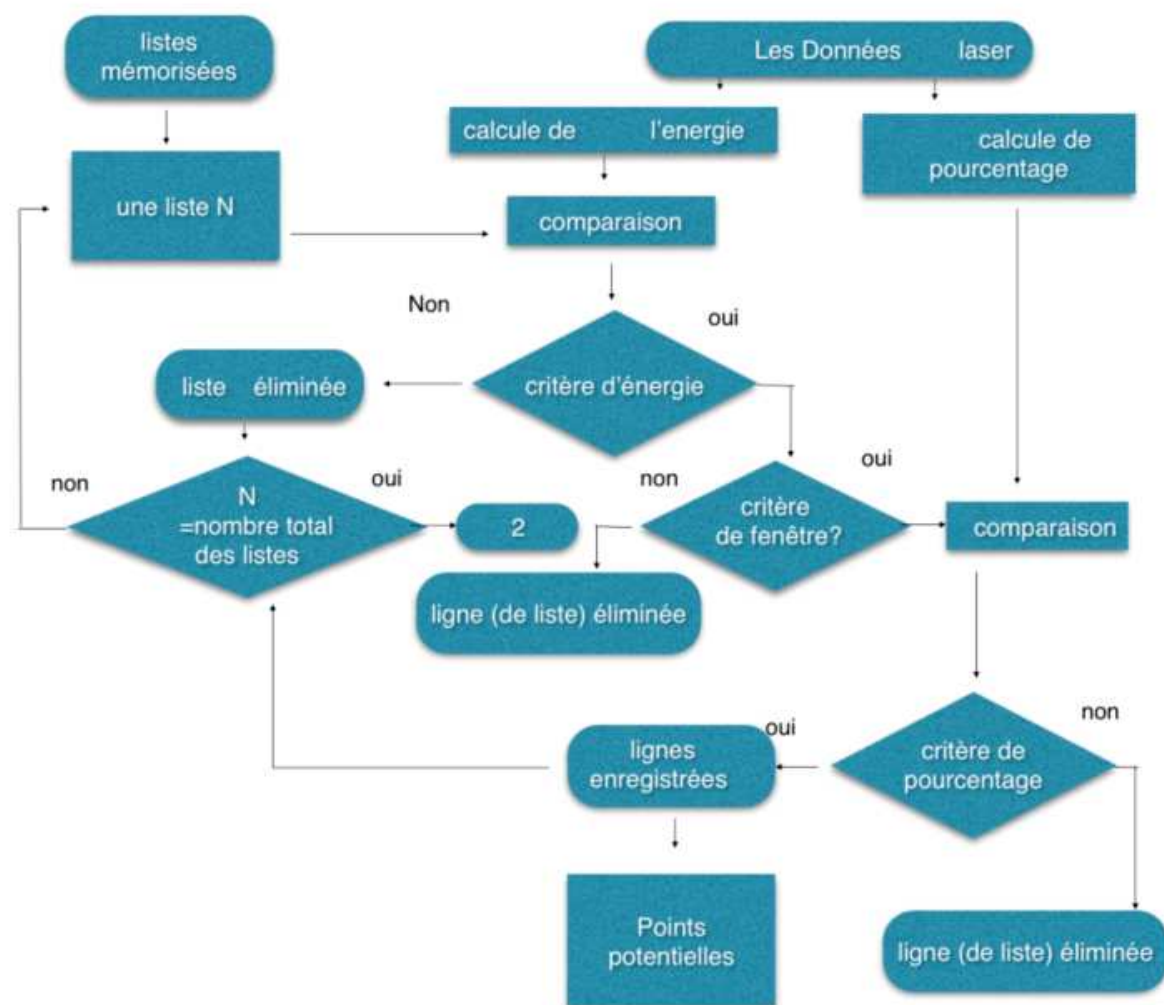


FIGURE 5.12 – Deuxième partie de l’algorithme de l’algorithme LZA (figure tirée de [Mou14])

Tableau 5.11 – Tableau récapitulatif des consommations pour les différents OIT (Table 3.1)

	Capteurs			Actionneurs			Batterie robot	Batterie PC												
	US	LAS	KIN	Arrêt	MCC (0.5 m/s)	MCC (0.1 m/s)	P_{BR} (W)	$P_{PC\ kinect}$	$P_{PC\ laser}$	$P_{watt} + Robot$	P_{CR1}	P_{CR2}	P_{ecran}	P_{Proc}	P_{PC} (W)	% P_{ecran}	$P_{totale} = P_{BR} + P_{PC}$	% P_{BR}	% P_{PC}	ΔP
<i>TV</i>	0	0	1	0	0	1	7,52	1,20	0,00	0,57	0,09	0,41	2,69	13,38	15,64	14,68	23,16	32,47	67,53	35,06
<i>S_</i> (suivi de chemin)	0	0	0	0	0	1	4,70	0,00	0,00	0,57	0,09	0,09	2,69	11,00	11,74	18,65	16,44	28,60	71,40	42,81
	0	0	0	0	1	0	10,12	0,00	0,00	0,57	0,09	0,09	2,69	11,75	12,49	17,72	22,61	44,75	55,25	10,49
<i>AS</i> (Avance Simple)	1	2	1	0	1	0	17,66	1,20	0,80	0,57	0,72	0,41	2,69	13,33	17,02	13,65	34,68	50,92	49,08	1,83
	0	2	1	0	1	0	17,62	1,20	0,80	0,57	0,72	0,41	2,69	13,27	16,96	13,69	34,58	50,95	49,05	1,90
	1	1	1	0	1	0	15,32	1,20	0,40	0,57	0,41	0,41	2,69	13,34	16,31	14,16	31,63	48,43	51,57	3,14
	0	1	1	0	1	0	15,28	1,20	0,40	0,57	0,41	0,41	2,69	13,28	16,25	14,20	31,53	48,46	51,54	3,08
	1	2	0	0	1	0	14,84	0,00	0,80	0,57	0,72	0,09	2,69	11,54	13,71	16,40	28,55	51,97	48,03	3,94
	0	2	0	0	1	0	14,80	0,00	0,80	0,57	0,72	0,09	2,69	11,48	13,65	16,46	28,45	52,02	47,98	4,04
	1	0	1	0	1	0	12,98	1,20	0,00	0,57	0,09	0,41	2,69	12,78	15,04	15,17	28,02	46,32	53,68	7,37
	0	0	1	0	1	0	12,94	1,20	0,00	0,57	0,09	0,41	2,69	12,72	14,98	15,22	27,92	46,34	53,66	7,31
	1	1	0	0	1	0	12,50	0,00	0,40	0,57	0,41	0,09	2,69	10,87	12,33	17,91	24,83	50,33	49,67	0,67
	0	1	0	0	1	0	12,46	0,00	0,40	0,57	0,41	0,09	2,69	10,81	12,27	17,98	24,73	50,38	49,62	0,76
	1	0	0	0	1	0	10,16	0,00	0,00	0,57	0,09	0,09	2,69	11,43	12,17	18,10	22,33	45,49	54,51	9,02
	0	0	0	0	1	0	10,12	0,00	0,00	0,57	0,09	0,09	2,69	11,37	12,11	18,18	22,23	45,52	54,48	8,96
<i>C_</i> (centrage)	1	2	1	0	1	0	17,66	1,20	0,80	0,57	0,72	0,41	2,69	14,15	17,84	13,10	35,50	49,74	50,26	0,52
	0	2	0	0	1	0	14,80	0,00	0,80	0,57	0,72	0,09	2,69	12,30	14,47	15,68	29,27	50,56	49,44	1,12
	1	2	0	0	1	0	14,84	0,00	0,80	0,57	0,72	0,09	2,69	12,36	14,53	15,62	29,37	50,52	49,48	1,04
	0	2	1	0	1	0	17,62	1,20	0,80	0,57	0,72	0,41	2,69	14,09	17,78	13,14	35,40	49,77	50,23	0,46
<i>LQR</i>	0	0	1	0	1	0	12,94	1,20	0,00	0,57	0,09	0,41	2,69	13,10	15,36	14,90	28,30	45,72	54,28	8,56
	1	1	1	0	1	0	15,32	1,20	0,40	0,57	0,41	0,41	2,69	13,72	16,69	13,88	32,01	47,85	52,15	4,29
	1	2	1	0	1	0	17,66	1,20	0,80	0,57	0,72	0,41	2,69	13,27	16,96	13,69	34,62	51,00	49,00	2,01
	0	1	1	0	1	0	15,28	1,20	0,40	0,57	0,41	0,41	2,69	13,66	16,63	13,92	31,91	47,88	52,12	4,24
	0	2	1	0	1	0	17,62	1,20	0,80	0,57	0,72	0,41	2,69	13,21	16,90	13,73	34,52	51,04	48,96	2,08
<i>LCH</i>	0	2	0	0	1	0	14,80	0,00	0,80	0,57	0,72	0,09	2,69	13,90	16,37	16,43	31,17	47,48	52,52	5,04
<i>DV</i>	0	0	1	1	0	0	5,49	1,20	0,00	0,57	0,09	0,41	2,69	11,74	14,00	16,12	19,49	28,17	71,83	43,66

Annexe B

Tableau 5.12 – $RSMD_1$ pour la mission simulée

Objectif ($x_d - x_f$)		O_5 (62 - 93.5)			O_6 (93.5)	O_7 (93.5)	O_8 (93.5)	O_9 (93.5 - 187)			
Tâche		DV LN			TV LN	AEIG	TV LN	DV LN			
Activité		A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
I	x_{A_i}	62	63.5	91.5	93.5	93.5	93.5	93.5	123.5	146	156
	dis_{A_i}	1.5	28	2	0	0	0	30	22.5	10	31
	dur_{A_i}	-	-	-	10	1	10	-	-	-	-
Liste des AI (Table 4.9)		1	1					1	1		1
		2	2					2	2		2
		3	3					3	3		3
		4	4					4	4		4
		5	5					5	5		5
		6	6					6	6	1	6
		7	7					7	7	2	7
		8	8	8				8	8	4	8
		9	9	9				9	9	7	9
		10	10	10	22		22	10	10	8	10
		11	11	11	23	25	23	11	11	9	11
		12	12	12	24		24	12	12	11	12
		13	13	13				13	13	14	13
		14	14	14				14	14	15	14
		15	15					15	15	16	15
		16	16					16	16	18	16
		17	17					17	17	21	17
		18	18					18	18		18
		19	19					19	19		19
		20	20					20	20		20
		21	21					21	21		21

Tableau 5.13 – $RSMD_2$ pour la mission simulée

Objectif ($x_d - x_f$)		O_9 (118 - 187)			
Tâche		DV LN			
Activité		A_{14}	A_{15}	A_{16}	A_{17}
I	x_{A_i}	118	123.5	146	156
	dis_{A_i}	5.5	22.5	10	31
	dur_{A_i}	-	-	-	-
Liste des AI (Table 4.9)		1	1		1
		2	2		2
		3	3		3
		4	4	1	4
		5	5	2	5
		6	6	4	6
		7	7	7	7
		15	15	15	15
		16	16	16	16
		17	17	18	17
		18	18	21	18
		19	19		19
		20	20		20
		21	21		21

Tableau 5.14 – $RSMD_3$ pour la mission simulée

Objectif ($x_d - x_f$)		O_9 (128 - 187)		
Tâche		DV LN		
Activité		A_{15}	A_{16}	A_{17}
I	x_{A_i}	128	146	156
	dis_{A_i}	18	10	31
	dur_{A_i}	-	-	-
Liste des AI (Table 4.9)		1		1
		2		2
		3		3
		4	1	4
		5	2	5
		6	4	6
		7	7	7
		15	15	15
		16	16	16
		17	18	17
		18	21	18
		19		19
		20		20
		21		21

Tableau 5.15 – $RSMD_4$ pour la mission simulée

Objectif ($x_d - x_f$)		O_9 (142 - 187)		
Tâche		DV LN		
Activité		A_{15}	A_{16}	A_{17}
I	x_{A_i}	142	146	156
	dis_{A_i}	4	10	31
	dur_{A_i}	-	-	-
Liste des AI (Table 4.9)		1		1
		2	1	2
		3	2	3
		15	15	15
		16	16	16
		17		17

Tableau 5.16 – $RSMD_1$ pour la mission réelle

Objectif ($x_d - x_f$)		O_5 (63 - 93.5)			O_6 (93.5)	O_7 (93.5)	O_8 (93.5)	O_9 (93.5 - 187)			
Tâche		DV LN			TV LN	AEIG	TV LN	DV LN			
Activité		A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
I	x_{A_i}	63	63.5	91.5	93.5	93.5	93.5	93.5	123.5	146	156
	dis_{A_i}	0.5	28	2	0	0	0	30	22.5	10	31
	dur_{A_i}	-	-	-	10	1	10	-	-	-	-
Liste des AI (Table 4.9)		1	1					1	1		1
		2	2					2	2		2
		3	3					3	3		3
		4	4					4	4		4
		5	5					5	5		5
		6	6					6	6	1	6
		7	7					7	7	2	7
		8	8	8				8	8	4	8
		9	9	9				9	9	7	9
		10	10	10	22		22	10	10	8	10
		11	11	11	23	25	23	11	11	9	11
		12	12	12	24		24	12	12	11	12
		13	13	13				13	13	14	13
		14	14	14				14	14	15	14
		15	15					15	15	16	15
		16	16					16	16	18	16
		17	17					17	17	21	17
		18	18					18	18		18
		19	19					19	19		19
		20	20					20	20		20
		21	21					21	21		21

Tableau 5.17 – $RSMD_2$ pour la mission réelle

Objectif ($x_d - x_f$)		O_9 (116 - 187)			
Tâche		DV LN			
Activité		A_{14}	A_{15}	A_{16}	A_{17}
I	x_{A_i}	116	123.5	146	156
	dis_{A_i}	7.5	22.5	10	31
	dur_{A_i}	-	-	-	-
Liste des AI (Table 4.9)		1	1		1
		2	2		2
		3	3	1	3
		8	8	2	8
		9	9	8	9
		10	10	9	10
		15	15	15	15
		16	16	16	16
		17	17		17

Tableau 5.18 – $RSMD_3$ pour la mission réelle

Objectif ($x_d - x_f$)		O_9 (171 - 187)
Tâche		DV LN
Activité		A_{17}
I	x_{A_i}	171
	dis_{A_i}	16
	dur_{A_i}	-
Liste des AI (Table 4.9)		1
		2
		3
		8
		9
		10
		15
		16
		17

Publications et présentations de l'auteur

Travaux publiés :

1. "Fault Tolerant Autonomous Robots Using Mission Performance Guided Resources Allocation", **Lotfi Jaïem**, Lionel Lapierre, Karen Godary-Dejean and Didier Crestani. *In Proc. of SysTol 2016*, pp. 45-50, Barcelone, Spain.
2. "Autonomous mobile robot navigation based on integrated environment representation designed in dynamic environments", Faten Cherni, Maïssa Boujelben, **Lotfi Jaïem**, Yassine Boutereaa, Chokri Rekik and Nabil Derbel. *International Journal of Automation and Control*, Vol. 11, No. 1, pp.35-53, 2017.
3. "Towards a Fault Tolerant Control Architecture with Performances Guarantees for Mobile Robots", **Lotfi Jaïem**, Lionel Lapierre, Karen Godary-Dejean and Didier Crestani, *Control Architecture of Robots*, Paris, 2014.
4. "Toward Performance Guarantee for Autonomous Mobile Robotic Mission : An Approach for Hardware and Software Resources Management", **Lotfi Jaïem**, Lionel Lapierre, Karen Godary-Dejean and Didier Crestani. *In Proc. of TAROS 2016, LNAI 9716*, pp. 189-195, Sheffield, U.K., 2016.
5. "A Step Toward Mobile Robots Autonomy : Energy Estimation Models", **Lotfi Jaïem**, Sebastien Druon, Lionel Lapierre and Didier Crestani. *In : Proc. of TAROS 2016, LNAI 9716*, pp. 177-188, Sheffield, U.K., 2016.
6. "Performance Guided Approach for Hardware and Software Resources Management for Autonomous Mobile Robotic Mission" (Long paper version of 4), **Lotfi Jaïem**, Robin Passama, Lionel Lapierre, Karen Godary-Dejean and Didier Crestani. *SHARC'2016*, Brest, France, 2016.

Travaux en cours :

1. "Stochastic Chromo-Localization LZA : a novel algorithm for mobile robot localization", René Zapata, Lei Zhang and **Lotfi Jaïem**, ROBOTICA.
2. "Energy consumption of control schemes for Pioneer 3DX mobile robot : models and evaluation", **Lotfi Jaïem**, Didier Crestani, Lionel Lapierre and Sebastien Druon, International Journal of Robotics Research.
3. "Energy Based Resources Allocation Using Performance Management for Autonomous Mobile Robotic Mission", **Lotfi Jaïem**, Lionel Lapierre, Robin Passama and Didier Crestani, ICRA 2017, Singapore.
4. "An Approach for Performance Management of Autonomous Mobile Robotic Mission", **Lotfi Jaïem**, Lionel Lapierre, Karen Godary-Dejean and Didier Crestani. IFAC'2017, Toulouse, France.

Bibliographie

- [AD10] David ASIKIN et John M DOLAN. “Reliability impact on planetary robotic missions”. In : *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE. 2010, p. 4095–4100.
- [Aka+10] Otar AKANYETI et al. “An application of Lyapunov stability analysis to improve the performance of NARMAX models”. In : *Rob. Auton. Syst.* 58.3 (2010), p. 229–238. ISSN : 09218890. DOI : [10 . 1016 / j . robot . 2009 . 11 . 001](https://doi.org/10.1016/j.robot.2009.11.001). URL : [http : // linkinghub . elsevier . com / retrieve / pii / S0921889009001924](http://linkinghub.elsevier.com/retrieve/pii/S0921889009001924).
- [Ala+06] R. ALAMI et al. “Safe and dependable physical human-robot interaction in anthropic domains : State of the art and challenges”. In : *IEEE Int. Conf. Intell. Robot. Syst.* October (2006). ISSN : 21530866. DOI : [10 . 1109 / IROS . 2006 . 6936985](https://doi.org/10.1109/IROS.2006.6936985).
- [Alb02] James S ALBUS. *Metrics and performance measures for intelligent unmanned ground vehicles*. Rapp. tech. DTIC Document, 2002.
- [Alb91] James S ALBUS. “Outline for a theory of intelligence”. In : *Systems, Man and Cybernetics, IEEE Transactions on* 21.3 (1991), p. 473–509.
- [AP08] Jorge ANGELES et Frank C PARK. “Performance evaluation and design criteria”. In : *Springer Handbook of Robotics*. Springer, 2008, p. 229–244.
- [BA15] Justin M BRADLEY et Ella M ATKINS. “Optimization and control of cyber-physical vehicle systems”. In : *Sensors* 15.9 (2015), p. 23020–23049.
- [BB03] Raymond BITEAU et Stéphanie BITEAU. *La maîtrise des flux industriels*. Éditions d’Organisation, 2003.
- [BBD00] L BENINI, a BOGLIOLO et G DE MICHELI. “A Survey of Design Techniques for System-Level Dynamic Power Management”. In : *IEEE Trans. VLSI Syst.* 8.3 (2000), p. 299–316. ISSN : 1063-8210. DOI : [10 . 1109 / 92 . 845896](https://doi.org/10.1109/92.845896). URL : [files / 1908 / ASurveyofDesignTechniquesforSystem - Level . pdf \\$ \backslash \\$ files / 1909 / freeabs { _ } all . html](http://files/1908/ASurveyofDesignTechniquesforSystem-Level.pdf%5Cbackslash$files/1909/freeabs%5C_$all.html).

- [BCP95a] A BARILI, M CERESA et C PARISI. “Energy-saving motion control for an autonomous mobile robot”. In : *Industrial Electronics, 1995. ISIE'95., Proceedings of the IEEE International Symposium on*. T. 2. IEEE. 1995, p. 674–676.
- [BCP95b] a. BARILI, M. CERESA et C. PARISI. “Energy-saving motion control for an autonomous mobile robot”. In : *Proc. IEEE Int. Symp. Ind. Electron.* 2 (1995), p. 674–676. DOI : [10.1109/ISIE.1995.497266](https://doi.org/10.1109/ISIE.1995.497266).
- [Ben+05] M. BENNOUR et al. “Computer-aided decision for human task allocation with mono- and multi-performance evaluation”. In : *Int. J. Prod. Res.* 43.21 (2005), p. 4559–4588. ISSN : 0020-7543. DOI : [10.1080/00207540500124579](https://doi.org/10.1080/00207540500124579). URL : <http://dx.doi.org/10.1080/00207540500124579>.
- [Ben14] Aalae BENKI. “Méthodes efficaces de capture de front de pareto en conception mère : applications industrielles Méthodes efficaces de capture de front de pareto en conception mécanique multicritère . A pplications industrielles”. In : (2014).
- [Ber15] Laurent BERDEU. *Génération d'événement, IHM*. Rapp. tech. LIRMM, 2015.
- [Bes95] P.L. BESCOS. *Contrôle de gestion et management*. Collection entreprendre. Montchrestien, 1995. ISBN : 9782707606730. URL : <https://books.google.fr/books?id=8dEnAAAACAAJ>.
- [BF96] Johann BORENSTEIN et Liqiang FENG. “Measurement and correction of systematic odometry errors in mobile robots”. In : *IEEE Transactions on robotics and automation* 12.6 (1996), p. 869–880.
- [BFR14] Jenay M BEER, Arthur D FISK et Wendy A ROGERS. “Toward a Framework for Levels of Robot Autonomy in Human-Robot Interaction”. In : *J. Human-Robot Interact.* 3.2 (2014), p. 74. ISSN : 2163-0364. DOI : [10.5898/JHRI.3.2.Beer](https://doi.org/10.5898/JHRI.3.2.Beer). URL : <http://hri-journal.org/index.php/HRI/article/view/125>.
- [BH01] S BRAINOV et H HEXMOOR. “Quantifying relative autonomy in multiagent interaction, in'IJCAI Workshop on Autonomy Delegation and Control : Interacting with Intelligent Agents’”. In : (2001).
- [BH03a] Sviatoslav BRAYNOV et Henry HEXMOOR. “Agent Autonomy”. In : sous la dir. d'Henry HEXMOOR, Cristiano CASTELFRANCHI et Rino FALCONE. Boston, MA : Springer US, 2003. Chap. Quantifying Relative Autonomy in Multiagent Interaction, p. 55–73. ISBN : 978-1-4419-9198-0. DOI : [10.1007/978-1-4419-9198-0_4](https://doi.org/10.1007/978-1-4419-9198-0_4). URL : http://dx.doi.org/10.1007/978-1-4419-9198-0_4.

- [BH03b] Sviatoslav BRAYNOV et Henry HEXMOOR. “Quantifying relative autonomy in multiagent interaction”. In : *Agent Autonomy*. Springer, 2003, p. 55–73.
- [BHP08] F BONSIGNORIO, J HALLAM et AP del POBIL. “Good experimental methodology guidelines”. In : *Special Interest Group on Good Experimental Methodology in Robotics European Robotics Research Network (EURON), Tech. Rep* (2008).
- [Bit90] M. BITTON. *ECOGRAI : méthode de conception et d’implantation de systèmes de mesure de performances pour organisations industrielles*. 1990. URL : <https://books.google.fr/books?id=8iN0tgAACAAJ>.
- [BJ12] William Lloyd BIRCHER et Lizy K JOHN. “Complete system power estimation using processor performance events”. In : *IEEE Transactions on Computers* 61.4 (2012), p. 563–577.
- [BK05] Alan BOWLING et Oussama KHATIB. “The dynamic capability equations : a new tool for analyzing robotic manipulator performance”. In : *Robotics, IEEE Transactions on* 21.1 (2005), p. 115–123.
- [BM05] Jean-Pierre BRANS et Bertrand MARESCHAL. *PROMETHEE methods Multiple criteria decision analysis : state of the art surveys (pp. 163-186)*. 2005.
- [Bou95] A. BOURGUIGNON. *Peut-on définir la performance ?* Rapp. tech. 1995.
- [Bro+00] David M BROOKS et al. “Power-aware microarchitecture : Design and modeling challenges for next-generation microprocessors”. In : *IEEE Micro* 20.6 (2000), p. 26–44.
- [Bro+83] Roger W BROCKETT et al. “Asymptotic stability and feedback stabilization”. In : *Differential geometric control theory* 27.1 (1983), p. 181–191.
- [Bro01] Tyson R BROWNING. “Applying the design structure matrix to system decomposition and integration problems : a review and new directions”. In : *Engineering Management, IEEE Transactions on* 48.3 (2001), p. 292–306.
- [BXL06] Jeff BRATEMAN, Changjiu XIAN et Yung Hsiang LU. “Energy-efficient scheduling for autonomous mobile robots”. In : *IFIP VLSI-SoIC 2006 - IFIP WG 10.5 Int. Conf. Very Large Scale Integr. Syst.* (2006), p. 361–366. DOI : [10.1109/VLSISOC.2006.313262](https://doi.org/10.1109/VLSISOC.2006.313262).
- [Car05] G CARAMANNA. “Scientific Diving and ROV Techniques Applied to the Geomorphological and Hydrogeological Study of the World’s Deepest Karst Sinkhole,(Pozzo del Merro–Latium–Italy).” In : (2005).

- [Cas95] Cristiano CASTELFRANCHI. “Guarantees for Autonomy in Cognitive Agent Architecture”. In : *Intell. Agents* 890 (1995), p. 56–70. ISSN : 16113349. DOI : [10.1007/3-540-58855-8_3](https://doi.org/10.1007/3-540-58855-8_3). URL : http://link.springer.com/chapter/10.1007/3-540-58855-8_{_}3.
- [CFK16] Ryan CALO, A Michael FROOMKIN et Ian KERR. *Robot law*. Edward Elgar Publishing, 2016.
- [CG03] Jacob W CRANDALL et Michael A GOODRICH. “Measuring the intelligence of a robot and its interface”. In : *Performance Metrics for Intelligent Systems (PerMIS’03)* (2003).
- [CGDL15] Didier CRESTANI, Karen GODARY-DEJEAN et Lionel LAPIERRE. “Enhancing fault tolerance of autonomous mobile robots”. In : *Robotics and Autonomous Systems* 68 (2015), p. 140–155.
- [Chi+13] Abdelghani CHIBANI et al. “Ubiquitous robotics : Recent challenges and future trends”. In : *Robotics and Autonomous Systems* 61.11 (2013), p. 1162–1172.
- [Cho07] Caroline CHOPINAUD. “Contrôle de l’émergence de comportements dans les systèmes d’agents cognitifs autonomes”. Thèse de doct. Paris 6, 2007.
- [Clo02] Bt CLOUGH. “Metrics, schmetrics ! How the heck do you determine a UAV’s autonomy anyway ?” In : *Security* 990 (2002), p. 313–319. ISSN : 1048-776X. URL : <http://oai.dtic.mil/oai/oai?verb=getRecord{\&}metadataPrefix=html{\&}identifier=ADA515926>.
- [CLTC95] F CHENAVIER, I LECOEUR TAIBI et JL CROWLEY. “Estimation de la Position d’un Robot par Odométrie et Vision Monoculaire”. In : *TS. Traitement du signal* 12.3 (1995), p. 255–268.
- [CM05a] Jennifer CARLSON et Robin R MURPHY. “How UGVs physically fail in the field”. In : *IEEE Transactions on robotics* 21.3 (2005), p. 423–437.
- [CM05b] Jennifer CARLSON et Robin R. MURPHY. “How UGVs physically fail in the field”. In : *IEEE Trans. Robot.* 21.3 (2005), p. 423–437. ISSN : 15523098. DOI : [10.1109/TR0.2004.838027](https://doi.org/10.1109/TR0.2004.838027).
- [CNP13] Ahmed CHEMORI, Guilherme Sartori NATAL et François PIERROT. “Control of parallel robots : towards very high accelerations”. In : (2013), p. 8.
- [Com03] Autonomic COMPUTING. “An architectural blueprint for autonomic computing”. In : *IBM Publication* (2003).
- [CS11] Yann COLLETTE et Patrick SIARRY. *Optimisation multiobjectif : Algorithmes*. Editions Eyrolles, 2011.

- [CVO10] Nelson David Muñoz CEBALLOS, Jaime Alejandro VALENCIA et Nelson Londoño OSPINA. *Quantitative performance metrics for mobile robots navigation*. INTECH Open Access Publisher, 2010.
- [Dal03] André DALGALARRONDO. “A propos de l’autonomie des robots”. In : 350.0 (2003), p. 108.
- [DCC03] Nakju DOH, Howie CHOSET et Wan Kyun CHUNG. “Accurate relative localization using odometry”. In : *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*. T. 2. IEEE. 2003, p. 1606–1612.
- [DDDB15] Didier DANET, Ronan DOARÉ et Gérard DE BOISBOISSEL. “Drones et killer robots”. In : (2015).
- [DH13] Rached DHAOUADI et Ahmad Abu HATAB. “Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies : A unified framework”. In : *Advances in Robotics & Automation 2013* (2013).
- [DI08] Ezequiel A. DI PAOLO et Hiroyuki IIZUKA. “How (not) to model autonomous behaviour”. In : *BioSystems* 91.2 (2008), p. 409–423. ISSN : 03032647. DOI : [10.1016/j.biosystems.2007.05.016](https://doi.org/10.1016/j.biosystems.2007.05.016).
- [Dur11] Bastien DURAND. “Proposition d’une architecture de contrôle adaptative pour la tolérance aux fautes”. Thèse de doct. UNIVERSITÉ MONTPELLIER II, 2011.
- [EK99] M.R. ENDSLEY et David B. KABER. “Level of automation effects on performance , situation”. In : *Ergonomics* 42.3 (1999), p. 462–492.
- [Erm00] Jean-Louis ERMINE. “Les systèmes de connaissances”. In : (2000), p. 144.
- [Far09] Jean-Loup FARGES. “PROTEUS : Platform for RObotic Modelling and Transformation for End- Users and Scientific communities - 3rd version”. In : (2009).
- [Fis68] Peter C FISHBURN. “Utility theory”. In : *Management science* 14.5 (1968), p. 335–378.
- [FOR80] J.W. FORRESTER. *Principes des systemes*. Science des systèmes. Presses Universitaires de Lyon, 1980.
- [FVI07] Tom FROESE, Nathaniel VIRGO et Eduardo IZQUIERDO. “Autonomy : a review and a reappraisal”. In : *European Conference on Artificial Life*. Springer. 2007, p. 455–464.
- [Fül05] János FÜLÖP. “Introduction to decision making methods”. In : *BDEI-3 workshop, Washington*. Citeseer. 2005.

- [GA10] Ronald T GREEN et Ben ABBOTT. “Groundwater voyager”. In : *Technology Today* (2010), p. 2.
- [GA88] Clement GOSSELIN et Jorge ANGELES. “A new performance index for the kinematic optimization of robotic manipulators”. In : *Proc. 20th ASME Mechanisms Conf.* 1988, p. 441–447.
- [Gal] PM GALLOIS. “Entreprise industrielle et typologie”. In : *Synthèse des travaux de la commission pilotage et évaluation de la performance industrielle AFGI Paris 1992* ().
- [Gar+08] Marcus GARY et al. “3D mapping and characterization of sistema Zacatón from DEPTHX (DEep Phreatic THERmal eXplorer)”. In : ASCE. 2008.
- [GD93] M GRIFFIS et J DUFFY. “Global stiffness modeling of a class of simple compliant couplings”. In : *Mechanism and machine theory* 28.2 (1993), p. 207–224.
- [GK88] Timothy J GRAETTINGER et Bruch H KROGH. “The acceleration radius : a global performance measure for robotic manipulators”. In : *Robotics and Automation, IEEE Journal of* 4.1 (1988), p. 60–69.
- [Gof10] Matthew GOFF. “Characterization of Error Growth in Wheel-Odometry Based on Dead Reckoning in a Differential Drive Robot”. Mém.de mast. University of Floride, 2010.
- [Goo08] James R. GOODWIN. “A unified Design Framework for Mobile Robot Systems”. Thèse de doct. Bristol Institute of Technology, University of the West of England, 2008.
- [Gro15] Teunis Harry de GROOT. *Mission-driven Resource Management for Reconfigurable Sensing Systems*. 2015. ISBN : 9789461863911.
- [Gun+14] Volkan GUNES et al. “A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems.” In : *TIIS* 8.12 (2014), p. 4242–4268.
- [GW08] James Richard GOODWIN et Alan WINFIELD. *A unified design framework for mobile robot systems*. University of the West of England, Bristol, 2008.
- [HAK] Kristo HEERO, A AABLOO et M KRUUSMAA. “on the Utility of Exploration on Time-Critical Mobile Robot Missions”. In : *Future* October 2015 (). URL : http://lepo.it.da.ut.ee/~maarjakr/kristo{_}heero{_}ECMR05.pdf.
- [HAM09] Daniel HELMICK, Anelia ANGELOVA et Larry MATTHIES. “Terrain adaptive navigation for planetary rovers”. In : *Journal of Field Robotics* 26.4 (2009), p. 391–410.

- [HASH08] Sami HADDADIN, Alin ALBU-SCHÄFFER et Gerd HIRZINGER. “The role of the robot mass and velocity in physical human-robot interaction - Part I : Non-constrained blunt impacts”. In : *Proc. - IEEE Int. Conf. Robot. Autom.* 011838 (2008), p. 1331–1338. ISSN : 10504729. DOI : [10.1109/ROBOT.2008.4543388](https://doi.org/10.1109/ROBOT.2008.4543388).
- [HASH09] Sami HADDADIN, Alin ALBU-SCHÄFFER et Gerd HIRZINGER. “Requirements for safe robots : Measurements, analysis and new insights”. In : *The International Journal of Robotics Research* 28.11-12 (2009), p. 1507–1527.
- [Her+10] Adolfo HERNANDO et al. “A Model-Based Approach to the Autonomic Management of Mobile Robot Resources”. In : *Proceedings of International Conference on Adaptive and Self-Adaptive Systems and Applications, Lisbon, Portugal*. Citeseer. 2010.
- [Hex00] Henry HEXMOOR. “A cognitive model of situated autonomy”. In : *Pacific Rim International Conference on Artificial Intelligence*. Springer. 2000, p. 325–334.
- [HGH99] Jin-Kao HAO, Philippe GALINIER et Michel HABIB. “Métaheuristiques pour l’optimisation combinatoire et l’affectation sous contraintes”. In : *Revue d’intelligence artificielle* 13.2 (1999), p. 283–324.
- [HJI06] Lupin HENRI, Marsot JACQUES et INRS. “Sécurité des machines et des équipements de travail. Moyens de protection contre les risques mécaniques”. In : (2006).
- [HM99] João P HESPANHA et A Stephen MORSE. “Stabilization of nonholonomic integrators via logic-based switching”. In : *Automatica* 35.3 (1999), p. 385–393.
- [Hol+05] PJ HOLLIDAY et al. “Understanding and measuring powered wheelchair mobility and manoeuvrability. Part I. Reach in confined spaces”. In : *Disability and rehabilitation* 27.16 (2005), p. 939–949.
- [Hor01] Paul HORN. “Autonomic computing : IBM\’s Perspective on the State of Information Technology”. In : (2001).
- [IEC99] ISO IEC. “Safety aspects-Guidelines for their inclusion in standards”. In : (1999).
- [IIN03] Koji IKUTA, Hideki ISHII et Makoto NOKATA. “Safety Evaluation Method of Design and Control for Human-Care Robots”. In : *Int. J. Rob. Res.* 22.5 (2003), p. 281–297. ISSN : 0278-3649. DOI : [10.1177/0278364903022005001](https://doi.org/10.1177/0278364903022005001). URL : <http://ijr.sagepub.com/cgi/content/abstract/22/5/281>.

- [IL85] Kaoru ISHIKAWA et David J LU. *What is total quality control ? : the Japanese way*. T. 215. Prentice-Hall Englewood Cliffs, NJ, 1985.
- [Ing+07] Félix INGRAND et al. “Decisional autonomy of planetary rovers”. In : *Journal of Field Robotics* 24.7 (2007), p. 559–580.
- [Ins12] Carlos C INSAURRALDE. “Autonomic management for the next generation of autonomous underwater vehicles”. In : *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE. 2012, p. 1–8.
- [Ins13] Carlos C INSAURRALDE. “Autonomic Management Capabilities for Robotics and Automation”. In : *1st Global Virtual Conference, Electronics, Electrical Systems, Electrical Engineering*. T. 16. 2013, p. 518–523.
- [IPP10] Edouard IVANJKO, Toni PETRINIC et Ivan PETROVIC. “Modelling of mobile robot dynamics”. In : *7th EUROSIM Congress on Modelling and Simulation*. T. 2. 2010.
- [ISO11a] DIS ISO. “DIS 13482 : Robots and robotic devices-Safety requirements-Non-medical personal care robot”. In : *Geneva, Switzerland* (2011).
- [ISO11b] ISO ISO. “ISO 10218-1 : 2011 : Robots and robotic devices-Safety requirements for industrial robots-Part 1 : Robots”. In : *Geneva, Switzerland : International Organization for Standardization* (2011).
- [Jac90] J. H. JACOT. *A propos de l'évaluation économique des systèmes intégrés de production*. Ecosip, Gestion Industrielle et Mesure Economique. Economica, 1990.
- [Jai+16a] Lotfi JAIEM et al. “A Step Toward Mobile Robots Autonomy : Energy Estimation Models”. In : *Proc. of TAROS 2016, LNAI 9716* (2016), p. 177–188.
- [Jai+16b] Lotfi JAIEM et al. “Fault Tolerant Autonomous Robots Using Mission Performance Guided Resources Allocation”. In : *Proc. of SysTol 2016* (2016).
- [Jai+16c] Lotfi JAIEM et al. “Toward Performance Guarantee for Autonomous Mobile Robotic Mission : An Approach for Hardware and Software Resources Management”. In : *Proc. of TAROS 2016, LNAI 9716* (2016), p. 189–195.
- [Kar11] Stamatis KARNOUSKOS. “Cyber-physical systems in the smartgrid”. In : *2011 9th IEEE International Conference on Industrial Informatics*. IEEE. 2011, p. 20–23.
- [KB02] Robert KOLLMAN et John BETTEN. “Powering electronics from the USB port”. In : *Internet/ Available : <http://www.ti.com/lit/an/slyt118/slyt118.pdf>* (2002).

- [KC13] Petar KORMUSHEV et Darwin G CALDWELL. “Improving the energy efficiency of autonomous underwater vehicles by learning to model disturbances”. In : *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, p. 3885–3892.
- [KGV82] S KIRKPATRICK, CD GELLATT et MP VECCHI. “Optimization by simulated annealing, IBM Thomas J”. In : *Watson research Center, Yorktown Heights, NY* (1982).
- [KH82] Dieter KLEIN et Edward HANNAN. “An algorithm for the multiple objective integer linear programming problem”. In : *European Journal of Operational Research* 9.4 (1982), p. 378–385.
- [Kim+14] Minjoong KIM et al. “A Simple Model for Estimating Power Consumption of a Multicore Server System”. In : *International Journal of Multimedia and Ubiquitous Engineering* 9.2 (2014), p. 153–160.
- [KK08a] Chong Hui KIM et Byung Kook KIM. “Minimum-Energy Motion Planning for Differential-Driven Wheeled Mobile Robots”. In : *Motion Plan.* (2008), p. 193–226.
- [KK08b] Hongjun KIM et Byung Kook KIM. “Minimum-energy translational trajectory planning for battery-powered three-wheeled omni-directional mobile robots”. In : *2008 10th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2008* December (2008), p. 1730–1735. DOI : [10.1109/ICARCV.2008.4795788](https://doi.org/10.1109/ICARCV.2008.4795788).
- [KK12] Kyoung-Dae KIM et Panganamala R KUMAR. “Cyber–physical systems : A perspective at the centennial”. In : *Proceedings of the IEEE* 100.Special Centennial Issue (2012), p. 1287–1308.
- [KN96] Robert S KAPLAN et David P NORTON. *The balanced scorecard : translating strategy into action*. Harvard Business Press, 1996.
- [KP07] Balajee KANNAN et Lynne E PARKER. “Metrics for quantifying system performance in intelligent, fault-tolerant multi-robot teams”. In : *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE. 2007, p. 951–958.
- [KPK99] Isaac KAMINER, Antonio PASCOAL et Wei KANG. “Integrated vision/inertial navigation system design using nonlinear filtering”. In : *American Control Conference, 1999. Proceedings of the 1999*. T. 3. IEEE. 1999, p. 1910–1914.
- [Kui06] Benjamin KUIPERS. “Building and evaluating an intelligent wheelchair”. In : (2006).

- [LAS15] Adrien LASBOUYGUES. “Exploration robotique de l ’ environnement aquatique : les modèles au cœur du contrôle”. Thèse de doct. Université de Montpellier, 2015.
- [LC07] François LEGRAS et Gilles COPPIN. “Autonomy spectrum for a multiple UAVs system”. In : *COGIS’07-COgnitive systems with Interactive Sensors* (2007).
- [LE95] M G LIPSETT et P ENG. “Robot Looseness Fault Diagnosis by”. In : (1995).
- [Lee+12a] Insup LEE et al. “Challenges and research directions in medical cyber–physical systems”. In : *Proceedings of the IEEE* 100.1 (2012), p. 75–90.
- [Lee+12b] Yu Huei LEE et al. “A near-optimum dynamic voltage scaling (DVS) in 65-nm energy-efficient power management with frequency-based control (FBC) for SoC system”. In : *IEEE J. Solid-State Circuits* 47.11 (2012), p. 2563–2575. ISSN : 00189200. DOI : [10.1109/JSSC.2012.2211671](https://doi.org/10.1109/JSSC.2012.2211671).
- [Lee08] Edward A LEE. “Cyber physical systems : Design challenges”. In : *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE. 2008, p. 363–369.
- [LL15] Jonathan R LESAGE et Raul G LONGORIA. “Mission Feasibility Assessment for Mobile Robotic Systems Operating in Stochastic Environments”. In : *Journal of Dynamic Systems, Measurement, and Control* 137.3 (2015), p. 031009.
- [Lo13] Mambaye LO. “Contribution à l’évaluation d’architectures en Ingénierie Système : application en conception de systèmes mécatroniques Mambaye Lo Spécialité : Génie Informatique , Automatique Présentée par Mambaye LÔ Contribution à l’évaluation d ”. In : (2013).
- [LOR96] Philippe LORINO. *Méthodes et pratiques de la performance*. Les éditions d’organisation, 1996.
- [LSP06] Lionel LAPIERRE, D SOETANTO et Antonio PASCOAL. “Nonsingular path following control of a unicycle in the presence of parametric modelling uncertainties”. In : *International Journal of Robust and Nonlinear Control* 16.10 (2006), p. 485–504.
- [LT06] Alexandre LAMPE et Catherine TESSIER. “Méthodologie d ’ évaluation du degré d ’ autonomie d ’ un robot mobile terrestre”. In : (2006).
- [Lyo+12] Damian M LYONS et al. “Designing autonomous robot missions with performance guarantees”. In : *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, p. 2583–2590.

- [LZ12] Lionel LAPIERRE et Rene ZAPATA. “A guaranteed obstacle avoidance guidance system the safe maneuvering zone”. In : *Auton. Robots* 32 (2012), p. 177–187. ISSN : 09295593. DOI : [10.1007/s10514-011-9269-5](https://doi.org/10.1007/s10514-011-9269-5).
- [Man+14] Maszatul M MANSOR et al. “Real-Time Improved Power Management for Autonomous Systems”. In : *IFAC World Congr. 2014* (2014), p. 2634–2639. ISSN : 14746670. DOI : [10.3182/20140824-6-ZA-1003.00854](https://doi.org/10.3182/20140824-6-ZA-1003.00854).
- [MBC01] Chantal MARTIN-BONNEFOUS et Alain COURTOIS. *Indicateurs de performance*. Hermes Science Publications, 2001.
- [Med11] Emmanuel MEDERNACH. “Université Blaise Pascal - Clermont-Ferrand II Ecole Doctorale Sciences Pour l ’ Ingénieur de Clermont-Ferrand Docteur d ’ Université”. In : (2011).
- [Mei+04] Yongguo MEI et al. “Energy-efficient motion planning for mobile robots”. In : *Robot. Autom. ...* (2004), p. 4344–4349. DOI : [10.1109/ROBOT.2004.1302401](https://doi.org/10.1109/ROBOT.2004.1302401). URL : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1302401&\backslashhttp://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1302401.
- [Mei+05a] Yongguo MEI et al. “Deployment strategy for mobile robots with energy and timing constraints”. In : *Proc. - IEEE Int. Conf. Robot. Autom.* 2005.3 (2005), p. 2816–2821. ISSN : 10504729. DOI : [10.1109/ROBOT.2005.1570540](https://doi.org/10.1109/ROBOT.2005.1570540).
- [Mei+05b] Yongguo Mei Yongguo MEI et al. “A case study of mobile robot’s energy consumption and conservation techniques”. In : *ICAR ’05. Proceedings., 12th Int. Conf. Adv. Robot. 2005.* (2005), p. 492–497. DOI : [10.1109/ICAR.2005.1507454](https://doi.org/10.1109/ICAR.2005.1507454).
- [Mei01] Leonard MEIROVITCH. “Fundamentals of vibrations”. In : (2001).
- [Met+53] Nicholas METROPOLIS et al. “Equation of state calculations by fast computing machines”. In : *The journal of chemical physics* 21.6 (1953), p. 1087–1092.
- [Mic+02] S MICHAUD et al. “SOLERO : SOLAR-POWERED EXPLORATION ROVER”. In : 1 (2002), p. 1–8.
- [MM05] R MADHAVAN et E MESSINA. “Performance evaluation of temporal range registration for autonomous vehicle navigation”. In : *Integr. Comput. Aided. Eng.* 12 (2005), p. 291–303. ISSN : 10692509.
- [Mm12] Amina MEKKI-MOKHTAR. ““ SE”. In : (2012).

- [Mon+10] Luis MONTESANO et al. "Towards an intelligent wheelchair system for users with cerebral palsy". In : *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 18.2 (2010), p. 193–202.
- [Mor98] C MORLEY. "LA GESTION DES RISQUES DANS LES PROJETS SYSTEMES D'INFORMATION". In : *Travail et méthodes* 543 (1998), p. 5–12.
- [Mot77] DC MOTORS. "Speed controls, servo systems : An engineering handbook". In : *Electro-Craft Corporation* (1977).
- [Mou14] Aliaa MOUALLA. *Mise en oeuvre et expérimentation d'une méthode de localisation d'un robot mobile*. Rapp. tech. LIRMM, 2014.
- [MPM10] Raj MADHAVAN, Angel del POBIL et Elena MESSINA. "Performance Evaluation and Benchmarking of Robotic and Automation Systems [TC Spotlight]". In : *IEEE Robotics & Automation Magazine* 1.17 (2010), p. 120–122.
- [Mur04] Richard MURCH. *Autonomic computing*. Ibm Press, 2004.
- [Net+11] a. Miranda NETO et al. "Real-time dynamic power management based on Pearson's Correlation Coefficient". In : *2011 15th Int. Conf. Adv. Robot.* (2011), p. 304–309. DOI : [10.1109/ICAR.2011.6088627](https://doi.org/10.1109/ICAR.2011.6088627).
- [Oga+06] Kazuya OGAWA et al. "Development of the robot power management system adapting to tasks and environments-The design guideline of the power control system applied to the distributed control robot". In : *2006 SICE-ICASE Int. Jt. Conf.* (2006), p. 2042–2046. DOI : [10.1109/SICE.2006.315489](https://doi.org/10.1109/SICE.2006.315489).
- [Ota09] Jun OTA. "Goal state optimization algorithm considering computational resource constraints and uncertainty in task execution time". In : *Rob. Auton. Syst.* 57.4 (2009), p. 403–410. ISSN : 09218890. DOI : [10.1016/j.robot.2008.07.002](https://doi.org/10.1016/j.robot.2008.07.002).
- [OZ00] Seungjun OH et Alexander ZELINSKY. "Autonomous battery recharging for indoor mobile robots". In : *Proc. Aust. Conf. Robot. Autom.* (2000), p. 1–6. URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.9145{\&}rep=rep1{\&}type=pdf>.
- [PA07] Robin PASSAMA et David ANDREU. "Towards a Language for Understanding Architectural Choices in Robotics". In : *ICRA'07 : Workshop Software Development and Integration in Robotics" Understanding Robot Software Architectures"*. 2007, CD-ROM.
- [Pag16] De Daniel PAGLIÉRO. "La sécurité en robotique La sécurité en robotique". In : 33.0 (2016).

- [Pal+10] T. PALLEJA et al. "Modeling floor-cleaning coverage performances of some domestic mobile robots in a reduced scenario". In : *Rob. Auton. Syst.* 58.1 (2010), p. 37–45. ISSN : 09218890. DOI : [10.1016/j.robot.2009.07.030](https://doi.org/10.1016/j.robot.2009.07.030). URL : <http://dx.doi.org/10.1016/j.robot.2009.07.030>.
- [Par+14] R PARASURAMAN et al. "Model based on-line energy prediction system for semi-autonomous mobile robots". In : *5th Int. Conf. Intell. Syst. Model. Simul.* (2014), p. 27–29.
- [Pas10] Robin PASSAMA. "ContrACT : une méthodologie de conception et de développement d ' architectures de contrôle de robots(Rapport LIRMM n° : RR-10025)". In : (2010), p. 1–45.
- [PBR93] Jean-Charles POMEROL et Sergio BARBA-ROMERO. *Choix multicritère dans l'entreprise : principes et pratique*. Hermes, 1993.
- [PCW13] Charles PIPPIN, Henrik CHRISTENSEN et Lora WEISS. "Performance based task assignment in multi-robot patrolling". In : *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM. 2013, p. 70–76.
- [Pis95] David PISINGER. "Algorithms for knapsack problems". In : (1995).
- [Poo+09] Radha POOVENDRAN et al. "A Community Report of the 2008 High Confidence Transportation Cyber-Physical Systems (HCTCPS) Workshop". In : (2009).
- [PR97] Raja PARASURAMAN et Victor RILEY. "Humans and Automation : Use , Misuse , Disuse , Abuse". In : *Hum. Factors J. Hum. Factors Ergon. Soc.* 39.2 (1997), p. 230–253. ISSN : 00187208. DOI : [10.1518/001872097778543886](https://doi.org/10.1518/001872097778543886).
- [Pre03] Kit PRESS. "Mars Exploration Rover Launches". In : *Management* June (2003).
- [Pru02] Jean PRUVOST. "Pierre Larousse : du grand dictionnaire au petit Larousse". In : (2002).
- [PSW00] R PARASURAMAN, T B SHERIDAN et C D WICKENS. "A model for types and levels of human interaction with automation." In : *IEEE Trans. Syst. Man. Cybern. A Syst. Hum.* 30.3 (2000), p. 286–97. ISSN : 1083-4427. DOI : [10.1109/3468.844354](https://doi.org/10.1109/3468.844354). URL : <http://www.ncbi.nlm.nih.gov/pubmed/11760769>.
- [Rav99] David RAVIART. "Proposition d'une méthodologie pour l'aide à l'évaluation des performances physico-économiques des systèmes de production dans une approche concourante". Thèse de doct. 1999.
- [Riv87] Eugene I RIVIN. *Mechanical design of robots*. McGraw-Hill, Inc., 1987.

- [RM06] I RANÓ et J MINGUEZ. “Steps towards the automatic evaluation of robot obstacle avoidance algorithms”. In : *Proc. of workshop of benchmarking in robotics, in the IEEE/RSJ int. conf. on intelligent robots and systems (IROS)*. 2006.
- [Roh+10] F. ROHRMULLER et al. “Interconnected performance optimization in complex robotic systems”. In : *Intell. Robot. Syst. (IROS), 2010 IEEE/RSJ Int. Conf. on. IEEE* (2010), p. 4113–4118. DOI : [10.1109/IROS.2010.5649728](https://doi.org/10.1109/IROS.2010.5649728). URL : http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=5649728.
- [Roy68] Bernard ROY. “Classement et choix en présence de points de vue multiples”. In : *Revue française d’automatique, d’informatique et de recherche opérationnelle. Recherche opérationnelle* 2.1 (1968), p. 57–75.
- [RP03] Peng RONG et Massoud PEDRAM. “An analytical model for predicting the remaining battery capacity of lithium-ion batteries”. In : *Proc. -Design, Autom. Test Eur. DATE* (2003), p. 1148–1149. ISSN : 15301591. DOI : [10.1109/DATE.2003.1253775](https://doi.org/10.1109/DATE.2003.1253775).
- [Saa08] Thomas L SAATY. “Decision making with the analytic hierarchy process”. In : *International journal of services sciences* 1.1 (2008), p. 83–98.
- [SAM05] Craig SCHLENOFF, Jerome AJOT et Raj MADHAVAN. “Performance evaluation of autonomous vehicle navigation in dynamic , on-road environments”. In : *Integr. Comput. Aided. Eng.* 12 (2005), p. 263–277. ISSN : 10692509.
- [SC82] J Kenneth SALISBURY et John J CRAIG. “Articulated hands force control and kinematic issues”. In : *The International journal of Robotics research* 1.1 (1982), p. 4–17.
- [Sch+09a] Debra SCHRECKENGHOST et al. “Measuring robot performance in real-time for NASA robotic reconnaissance operations”. In : *Proc. 9th Work. Perform. Metrics Intell. Syst. - Permis ’09* (2009), p. 194. DOI : [10.1145/1865909.1865950](https://doi.org/10.1145/1865909.1865950). URL : <http://dl.acm.org/citation.cfm?id=1865909.1865950>.
- [Sch+09b] Debra SCHRECKENGHOST et al. “Measuring robot performance in real-time for NASA robotic reconnaissance operations”. In : *Proceedings of the 9th Workshop on Performance Metrics for Intelligent Systems*. ACM. 2009, p. 194–202.
- [Sch+09c] Debra SCHRECKENGHOST et al. “Real-time assessment of robot performance during remote exploration operations”. In : *IEEE Aerosp. Conf. Proc.* (2009). ISSN : 1095323X. DOI : [10.1109/AERO.2009.4839310](https://doi.org/10.1109/AERO.2009.4839310).

- [Sch85] Alain SCHÄRLIG. *Décider sur plusieurs critères : panorama de l'aide à la décision multicritère*. T. 1. PPUR presses polytechniques, 1985.
- [Sel+06a] Brennan SELLNER et al. "Coordinated multiagent teams and sliding autonomy for large-scale assembly". In : *Proc. IEEE* 94 (2006), p. 1425–1443. ISSN : 00189219. DOI : [10.1109/JPROC.2006.876966](https://doi.org/10.1109/JPROC.2006.876966).
- [Sel+06b] Brennan SELLNER et al. "Coordinated multiagent teams and sliding autonomy for large-scale assembly". In : *Proc. IEEE* 94 (2006), p. 1425–1443. ISSN : 00189219. DOI : [10.1109/JPROC.2006.876966](https://doi.org/10.1109/JPROC.2006.876966).
- [Set12] ML SETO. "Application of On-Board Evolutionary Algorithms to Underwater Robots to Optimally Replan Missions with Energy Constraints". In : *Journal of Robotics* 2012 (2012).
- [Sie+03] Maarten SIERHUIS et al. "Modeling and Simulation for Mission Operations Work System Design". In : *J.Manage.Inf.Syst.* 19.4 (2003), p. 85–128. ISSN : 0742-1222.
- [Sim60] Herbert A SIMON. "The new science of management decision." In : (1960).
- [Sim96] HA SIMON. *The Sciences of the Artificial, the 3rd Edition*. 1996.
- [SJU12] Amir SADRPOUR, J JIN et A Galip ULSOY. "Mission energy prediction for unmanned ground vehicles". In : *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, p. 2229–2234.
- [SJU13a] Amir SADRPOUR, J JIN et a G ULSOY. "Experimental Validation of Mission Energy Prediction Model for Unmanned Ground Vehicles". In : *Proc. 2013 Am. Control Conf.* (2013), p. 5980–5985. ISSN : 07431619.
- [SJU13b] Amir SADRPOUR, Jionghua Judy JIN et a Galip ULSOY. "Mission Energy Prediction for Unmanned Ground Vehicles Using Real-time Measurements and Prior Knowledge". In : *J. F. Robot.* 30.3 (2013), p. 399–414. ISSN : 15564959. DOI : [10.1002/rob.21453](https://doi.org/10.1002/rob.21453). URL : <http://doi.wiley.com/10.1002/rob.21453>.
- [SM08] Victor SCHEINMAN et J Michael MCCARTHY. "Mechanisms and actuation". In : *Springer Handbook of Robotics*. Springer, 2008, p. 67–86.
- [SMF10] Debra SCHRECKENGHOST, Tod MILAM et Terrence FONG. "Measuring performance in real time during remote human-robot operations with adjustable autonomy". In : *IEEE Intell. Syst.* 25 (2010), p. 36–44. ISSN : 15411672. DOI : [10.1109/MIS.2010.126](https://doi.org/10.1109/MIS.2010.126).
- [SN04] Roland SIEGWART et IR NOURBAKHS. "Introduction to Autonomous Mobile Robots, 2004". In : *Bradford Book* (2004).

- [Soh+10] Ripduman SOHAN et al. “Characterizing 10 Gbps network interface energy consumption.” In : *LCN*. 2010, p. 268–271.
- [Soi] IEEE Robotics & Automation SOCIETY. *Performance Evaluation & Benchmarking of Robotic and Automation Systems*. URL : <http://www.ieee-ras.org/performance-evaluation>.
- [SPB02a] Richard C SIMPSON, Daniel POIROT et Francie BAXTER. “The Hephaestus smart wheelchair system”. In : *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 10.2 (2002), p. 118–122.
- [SPB02b] Richard C. SIMPSON, Daniel POIROT et Francie BAXTER. “The Hephaestus Smart Wheelchair System”. In : *IEEE Trans. Neural Syst. Rehabil. Eng.* 10.2 (2002), p. 118–122. ISSN : 15344320. DOI : [10.1109/TNSRE.2002.1031980](https://doi.org/10.1109/TNSRE.2002.1031980).
- [SSA13] Daniel SIMON, Ye-Qiong SONG et Christophe AUBRUN. *Co-design approaches to dependable networked control systems*. John Wiley & Sons, 2013.
- [SSP02] Eleftheria S SERGAKI, George S STAVRAKAKIS et Anastasios D POULIEZOS. “Optimal robot speed trajectory by minimization of the actuator motor electromechanical losses”. In : *Journal of Intelligent and Robotic Systems* 33.2 (2002), p. 187–207.
- [Ste46] SS STEFENS. “On the theory of scales of measurement”. In : *Science*. — January-june (1946).
- [Ste95] Luc STEELS. “When are robots intelligent autonomous agents?” In : *Rob. Auton. Syst.* 15.1-2 (1995), p. 3–9. ISSN : 09218890. DOI : [10.1016/0921-8890\(95\)00011-4](https://doi.org/10.1016/0921-8890(95)00011-4).
- [Suk97] G.S. SUKHATME. “On the evaluation of autonomous mobile robots”. In : May (1997). URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.17.899{\&}rep=rep1{\&}type=pdf>.
- [Sun09] Teruo SUNAGA. “Theory of an interval algebra and its application to numerical analysis”. In : *Japan Journal of Industrial and Applied Mathematics* 26.2 (2009), p. 125–143.
- [Tah03] Christian TAHON. *Evaluation des performances des systèmes de production*. Lavoisier, 2003.
- [TB06] Vincent T’KINDT et Jean-Charles BILLAUT. *Multicriteria scheduling : theory, models and algorithms*. Springer Science & Business Media, 2006.
- [TC 12] IEC TC 59. *Performance evaluation method of intelligent mobile robot platform for household and similar applications*. Rapp. tech. International Standards et Conformity, 2012, p. 1–25.

- [TKI11a] Pratap TOKEKAR, Nikhil KARNAD et Volkan ISLER. “Energy-optimal velocity profiles for car-like robots”. In : *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, p. 1457–1462.
- [TKI11b] Pratap TOKEKAR, Nikhil KARNAD et Volkan ISLER. “Energy-optimal velocity profiles for car-like robots”. In : *Proc. - IEEE Int. Conf. Robot. Autom.* 1 (2011), p. 1457–1462. ISSN : 10504729. DOI : [10.1109/ICRA.2011.5980374](https://doi.org/10.1109/ICRA.2011.5980374).
- [TKI14] Pratap TOKEKAR, Nikhil KARNAD et Volkan ISLER. “Energy-optimal trajectory planning for car-like robots”. In : *Autonomous Robots* 37.3 (2014), p. 279–300.
- [Tre+10] David TREJO et al. “Performance Evaluation for Autonomous Mobile Robots”. In : (2010), p. 365–371.
- [Tru+09] Walt TRUSZKOWSKI et al. *Autonomous and autonomic systems : With applications to NASA intelligent spacecraft operations and exploration systems*. Springer Science & Business Media, 2009.
- [Tun07] Edward TUNSTEL. “Operational performance metrics for mars exploration rovers”. In : *Journal of Field Robotics* 24.8-9 (2007), p. 651–670.
- [Ull02] D.G. ULLMAN. “The Ideal Engineering Decision Support System”. In : *Retrieved March 10 (2002)*, p. 2009. URL : <http://www.inta.gov.ar/bariloche/desarrollo/gesrural/trabajos/ssd/Bibliografia/ProcesoTomadedecisiones/TheIdealEngineeringDecisionSupportSystem.pdf>.
- [UPÇ12] A Galip ULSOY, Huei PENG et Melih ÇAKMAKCI. *Automotive control systems*. Cambridge University Press, 2012.
- [UTF95] Berthold ULUNGU, Jacques TEGHEM et Philippe FORTEMPS. “Heuristic for multi-objective combinatorial optimization problems by simulated annealing”. In : *MCDM : Theory and Applications 1995* (1995).
- [VB90] Hendrik VAN BRUSSEL. “Evaluation and testing of robots”. In : *CIRP Annals-Manufacturing Technology* 39.2 (1990), p. 657–664.
- [Wei+12] Hongxing WEI et al. “Staying-alive path planning with energy optimization for mobile robots”. In : *Expert Syst. Appl.* 39.3 (2012), p. 3559–3571. ISSN : 09574174. DOI : [10.1016/j.eswa.2011.09.046](https://doi.org/10.1016/j.eswa.2011.09.046). URL : <http://dx.doi.org/10.1016/j.eswa.2011.09.046>.
- [Woi14] Hans Christian WOITHE. “Design and implementation of an energy aware programming framework for autonomous underwater vehicles”. Thèse de doct. Citeseer, 2014.

- [Won+02] Sylvia C WONG et al. “Performance metrics for robot coverage tasks”. In : *Proceedings of Australasian Conference on Robotics and Automation*. T. 27. 2002, p. 29.
- [WV07] J WAWERLA et R T VAUGHAN. “Near-optimal mobile robot recharging with the rate-maximizing forager”. In : *Proc. 9th Eur. Conf. Adv. Artif. life* (2007), p. 776–785.
- [WV08] Jens WAWERLA et Richard T VAUGHAN. “w3 Optimal Robot Recharging Strategies For Time Discounted Labour”. In : *Artif. Life* 2003 (2008), p. 670–677.
- [Xu+04] Ruibin XU et al. “Practical PACE for embedded systems”. In : *Proceedings of the 4th ACM international conference on Embedded software*. ACM. 2004, p. 54–63.
- [ZCL04] René ZAPATA, Alessio CACITTI et Pascal LEPINAY. “DVZ-based collision avoidance control of non-holonomic mobile manipulators”. In : *Journal européen des systemes automatisés* 38.5 (2004), p. 559–588.
- [ZH07] Wei Zhang Wei ZHANG et Jianghai Hu Jianghai HU. “Low power management for autonomous mobile robots using optimal control”. In : *2007 46th IEEE Conf. Decis. Control* (2007), p. 5364–5369. ISSN : 0191-2216. DOI : [10.1109/CDC.2007.4434847](https://doi.org/10.1109/CDC.2007.4434847).
- [ZLH09a] Wei ZHANG, Yung-Hsing LU et Jianghai HU. “Optimal solutions to a class of power management problems in mobile robots”. In : *Automatica* 45.4 (2009), p. 989–996. ISSN : 00051098. DOI : [10.1016/j.automatica.2008.11.004](https://doi.org/10.1016/j.automatica.2008.11.004).
- [ZLH09b] Wei ZHANG, Yung-Hsing LU et Jianghai HU. “Optimal solutions to a class of power management problems in mobile robots”. In : *Automatica* 45.4 (2009), p. 989–996. ISSN : 00051098. DOI : [10.1016/j.automatica.2008.11.004](https://doi.org/10.1016/j.automatica.2008.11.004). URL : <http://dx.doi.org/10.1016/j.automatica.2008.11.004>.
- [ZLV07] Pawel ZEBROWSKI, Yaroslav LITUS et Richard VAUGHAN. “Energy-Efficient Robot Rendezvous”. In : *Fourth Can. Conf. Comput. Robot Vis. (CRV '07)* (2007), p. 139–148.
- [Ben00] Sami BEN MENA. “Introduction aux méthodes multicritères d’aide à la décision”. In : *Biotechnol. Agron. Soc. Environ.* 4.2 (2000), p. 83–93. ISSN : 17804507.

Contribution to robot autonomy : Toward performance guarantee for autonomous mobile robotics by hardware and software resources management

Abstract :

The performance is a multi-form concept widely defined in manufacturing robotics with specific environment conditions (static and perfectly known) and infinite energy). However, performance indicators proposed in mobile robotics are less widely accepted. We differentiate between main performance axes (energy, safety, localization and stability) and secondary performance axes (duration for example). In our work, we are interested on missions realized under duration, safety and energy performance constraints, in a known but dynamic environment. Applying the different constraints decomposes the mission into a sequence of activities realized under invariant constraints. Each one, can be realized by a set of robotic tasks (move, be located, analyze an image, etc.). These tasks can be implemented in various ways according to the different possibles actuators, sensors and algorithms configurations.

The adressed problem is the following : how to choose the hardware and software resources to use along a mission while satisfying the different performance constraints ? It is a multicriteria knapsack problem known to be NP-hard, where the complexity becomes very quickly unexplorable. To propose and guarantee an applicable solution under real-time constraints, we used an algorithm allowing to find a set of good solutions in few iterations.

The proposed resources management approach is implemented on a Pioneer-3DX robot using a control architecture based on the Middleware ContrACT. This approach has been validated on a patrolling mission travelling 200 m within the LIRMM laboratory during about 10 mn, to verify the state of valves. For the considered mission, the state space dimension is higher than 10^{14} . The hardware and software resources are dynamically and autonomously selected along the mission to satisfy the different performance constraints. If a resource becomes faulty and/or many obstacle avoidances occure and lead to performance drift, the developed approach finds on line a new resources allocation solution (if it exists). So this approach allows to enhance the fault tolerance of the robotic system.

Keywords : Mobile robotics, resources management, autonomy, decision, performance.

Contribution à l'autonomie des robots : Vers la garantie de performance en robotique mobile autonome par la gestion des ressources matérielles et logicielles

Résumé :

La performance est un concept multiforme largement décliné et défini en robotique de manipulation où il relève de contraintes spécifiques : environnement non dynamique de dimensions limitées, énergie infinie. Les indicateurs de performance proposés dans le cadre de missions robotiques terrestres sont moins largement acceptés. Pour notre part nous les déclinons en axes principaux : énergie, sécurité, localisation, stabilité, et secondaires : durée par exemple.

Dans le cadre de ce travail nous nous intéressons à des missions robotiques soumises à des contraintes de performance liées à la durée, la sécurité et l'énergie dans un environnement connu mais dynamique. La déclinaison des contraintes imposées à la mission et au robot permet de décomposer la mission en une suite d'activités aux contraintes invariantes. Chacune de ces activités pouvant être réalisée à l'aide d'un ensemble de tâches robotiques (se déplacer, se localiser, analyser une image, etc.) pouvant elles-mêmes être implémentées de différentes façons en fonction des actionneurs, capteurs ou algorithmes pouvant être utilisés.

Le problème adressé est le suivant : comment choisir les ressources matérielles et logicielles à utiliser tout au long de la mission de façon à vérifier les contraintes de performance imposées ? C'est un problème de sac à dos multicritères NP-Complet et l'espace de recherche devient très rapidement inexplorable. De façon à proposer et garantir une solution applicable sous des contraintes temps réel, nous faisons appel à un algorithme permettant de trouver un ensemble ordonné de bonnes solutions en temps linéaire.

L'approche de gestion de ressources proposée a été implémentée sur un robot Pioneer-3DX et une architecture de contrôle s'appuyant sur le *Middleware* ContrACT. Cette approche a été validée dans le cadre d'une mission de patrouille d'une longueur de 200 m et durant une dizaine de minutes, au sein du laboratoire LIRMM, pour vérifier l'état de vannes. Pour la mission choisie, l'espace d'états à considérer est supérieur à 10^{14} . Tout au long de la mission, les ressources matérielles et logicielles sont choisies dynamiquement et d'une façon autonome afin de satisfaire les contraintes de performance imposées. Si des ressources ne sont plus opérationnelles, ou si trop d'évitements d'obstacles non prévus sont effectués, l'approche développée est capable de trouver en ligne une nouvelle solution d'affectation des ressources vérifiant les contraintes de performance imposées, si elle existe. Ces travaux participant donc à l'accroissement du niveau de tolérance aux fautes du système robotisé.

Mots-clés : Robotique mobile, gestion ressources, autonomie, décision, performance.