

THÈSE DE DOCTORAT

de

l'Université d'Aix-Marseille

préparée au

Laboratoire d'Informatique et Systèmes
(LIS UMR 7020)

École Doctorale en Mathématiques et Informatique de Marseille
(ED 184)

Spécialité de doctorat : Diagnostic, Automatique, et Robotique Mobile

Par

Samia MELLAH

Diagnostic de défauts des robots mobiles à roues, et gestion du retard généré par les obstacles mobiles.

Soutenue publiquement, le 14 décembre 2020

Devant le jury composé de :

M. José RAGOT	Professeur émérite des Universités Université de Lorraine, CRAN UMR 7039	Rapporteur
M. Frédéric KRATZ	Professeur des Universités l'INSA Centre Val de Loire, PRISME EA 4229	Rapporteur
Mme Ouidad LABBANI- IGBIDA	Professeur des Universités Université de Limoges, XLIM UMR 7252	Examinaterice
Mme Agnès ROUSSY	Maître de conférences Mines Saint Etienne, LIMOS UMR 6158	Examinaterice
M. Mustapha OULADSINE	Professeur des Universités Aix-Marseille Marseille, LIS UMR 7020	Directeur de thèse
M. Guillaume GRATON	Maître de Conférences Ecole Centrale Marseille, LIS UMR 7020	Encadrant de thèse
M. El-Mostafa EL ADEL	Maître de Conférences Aix-Marseille Marseille, LIS UMR 7020	Encadrant de thèse
M. Alain PLANCHAIS	Ingénieur, Chef de projets STMicroelectronics, Rousset	Encadrant industriel

Samia MELLAH : *Diagnostic de défauts des robots mobiles à roues, et gestion du retard généré par les obstacles mobiles.*

Thèse de Doctorat

14 décembre 2020 Marseille, France



Cette œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 4.0 International.

ABSTRACT

The work presented in this thesis is a part of the European project *PRODUCTIVE4.0* in partnership with *STMicroelectronics* of Rousset, France. In order to optimize the production, *STMicroelectronics* integrates two types of wheeled mobile robots (unicycle mobile robots and omnidirectional mobile robots with 4 Mecanum wheels) in the fab (semiconductor manufacturing facility) to transport products between the various equipments in the production chain. In this thesis, we distinguish two major issues that can prevent the robots from well performing their task : the apparition of *faults* at the actuators or the sensors level, and the delay that can be generated by moving and unpredictable obstacles while navigating.

Firstly, we are interested in unicycle robots. We propose a Kalman observer-based approach to diagnose the actuators and sensors *faults*. Then, an approach based on the *faults* estimation and compensation allows to accommodate these faults. Secondly, we focus on omnidirectional robots and we propose methods based on observers (Kalman and unknown inputs) in order to diagnose the actuators and sensors *faults*. After that, the impact of the actuators *faults* on the robots behavior is studied. Finally, we deal with the problem linked to the obstacles generated delay by proposing a methodology based on the reference trajectories reconfiguration in order to compensate this delay.

RÉSUMÉ

Ce travail de thèse s'inscrit dans le cadre du projet Européen *PRODUCTIVE4.0* en partenariat avec *STMicroelectronics* de Rousset, France. Afin d'optimiser la production, *STMicroelectronics* intègre deux types de robots mobiles à roues (les robots unicycles et les robots omnidirectionnels à 4 roues Mecanum) dans la fab (installations de fabrication du semi-conducteur) pour le transport des produits entre les divers équipements de la chaîne de production. Dans le cadre des travaux de cette thèse, nous distinguons deux problématiques majeures pouvant empêcher les robots de bien effectuer leur tâche : l'apparition de *défauts* au niveau des actionneurs ou des capteurs, et le retard qui peut être généré par les obstacles mobiles et imprévisibles lors de la navigation des robots.

Nous nous intéressons dans un premier temps aux robots unicycles. Nous proposons une approche à base d'observateur de Kalman afin de diagnostiquer les *défauts* actionneurs et capteurs. Une approche basée sur l'estimation et la compensation des *défauts* permet ensuite de les accommoder. Dans un second temps, nous nous focalisons sur les robots omnidirectionnels et nous proposons des méthodes à base d'observateurs (de Kalman et à entrées inconnues) afin de diagnostiquer les *défauts* actionneurs et capteurs. Ensuite, l'impact des *défauts* actionneurs sur le comportement des robots est étudié. Enfin, nous traitons la problématique liée au retard généré par les obstacles en proposant une méthodologie basée sur la reconfiguration des trajectoires de référence afin de compenser ce retard.

TRAVAUX PUBLIÉS

- 2020 [1] **Samia MELLAH**, Guillaume GRATON, El Mostafa EL ADEL, Mustapha OULADSINE, and Alain PLANCHAIS. **Trajectory reconfiguration for time delay reduction in the case of unexpected obstacles : application to 4-mecanum wheeled mobile robots (4-MWMR) for industrial purposes.** In *IFAC World Congress 2020*, Berlin, Germany, July 2020.
- [2] **Samia MELLAH**, Guillaume GRATON, El Mostafa EL ADEL, Mustapha OULADSINE, and Alain PLANCHAIS. **4-mecanum wheeled mobile robot actuator fault detection & isolation using unknown input observer-based approach.** In *European Control Conference (ECC) 2020*, Saint Petersburg, Russia, May 2020.
- 2019 [3] **Samia MELLAH**, Guillaume GRATON, El Mostafa EL ADEL, Mustapha OULADSINE, and Alain PLANCHAIS. **Trajectory tracking and time delay management of 4-mecanum wheeled mobile robots (4-MWMR).** In *15th European Workshop on Advanced Control and Diagnosis, ACD 2019*, Bologna, Italy, November 2019.
- [4] **Samia MELLAH**, Guillaume GRATON, El Mostafa EL ADEL, Mustapha OULADSINE, and Alain PLANCHAIS. **Mobile robot additive fault diagnosis and accommodation.** In *International Conference on Systems and Control (ICSC) 2019*, Marrakesh, Morocco, October 2019.
- [5] **Samia MELLAH**, Guillaume GRATON, El Mostafa EL ADEL, Mustapha OULADSINE, and Alain PLANCHAIS. **Detection & isolation of sensor and actuator additive faults in a 4-mecanum wheeled mobile robot (4-MWMR).** In *International Conference on Control, Automation and Diagnosis (ICCAD) 2019*, Grenoble, France, July 2019.
- 2018 [6] **Samia MELLAH**, Guillaume GRATON, El Mostafa EL ADEL, Mustapha OULADSINE, and Alain PLANCHAIS. **On fault detection and isolation applied on unicycle mobile robot sensors and actuators.** In *International Conference on Systems and Control (ICSC) 2018*, Valencia, Spain, October 2018.
-

TRAVAUX SOUMIS

-
- 2021 [7] **Samia MELLAH**, Guillaume Graton, El Mostafa El Adel, Mustapha OULADSINE, and Alain PLANCHAIS. **Monitoring of actuator health state & study of the degradation impact on a 4-mecanum wheeled mobile robot behaviour**. In *American Control Conference (ACC) 2021*, New Orleans, Louisiana, USA, May 2021.
- [8] **Samia MELLAH**, Guillaume Graton, El Mostafa El Adel, Mustapha OULADSINE, and Alain PLANCHAIS. **Health state monitoring of 4-mecanum wheeled mobile robot actuators and its impact on the robot behaviour analysis**. In *Journal of Intelligent & Robotic Systems*.
- [9] **Samia MELLAH**, Guillaume Graton, El Mostafa El Adel, Mustapha OULADSINE, and Alain PLANCHAIS. **Trajectory reconfiguration for unexpected obstacles generated time delay reduction : application to 4-mecanum wheeled mobile robots for industrial purposes**. In *IEEE Transactions on Intelligent Transportation Systems*.
-

«*Un jour j'irai vivre en Théorie car en Théorie tout se passe bien.*»

—*Marc Levy*

REMERCIEMENT

Les travaux de cette thèse ont été réalisés au sein de l'équipe PECASE du Laboratoire d'Informatique et Systèmes à l'Université d'Aix-Marseille, où j'ai eu le privilège d'être entourée par des personnes généreuses et très agréables. Avant de commencer ce manuscrit, je souhaite adresser mes remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de cette thèse.

Avant tout, je tiens à remercier chaleureusement mon directeur de thèse M. Mustapha OULADSINE, professeur à l'université d'Aix-Marseille, de m'avoir accueillie au sein du laboratoire et d'avoir dirigé mes travaux de recherche. Je le remercie pour sa confiance, ses précieux conseils, ses orientations et sa rigueur scientifique.

J'exprime ma profonde gratitude à mon encadrant M. Guillaume GRATON, maître de conférences à Centrale Marseille, pour son encadrement, son aide, son soutien, son encouragement, sa disponibilité tout au long de ces années de recherche, et surtout pour sa gentillesse et ses précieuses qualités humaines.

Je tiens également à remercier vivement mon encadrant M. El Mostafa EL ADEL pour sa gentillesse, son encadrement, ses conseils, et sa disponibilité.

Je tiens également à témoigner ma reconnaissance à M. Alain PLANCHAIS, le chef du projet PRODUCTIVE4.0 chez STMicroelectronics pour sa collaboration et ses remarques pertinentes.

Mes remerciements s'adressent ensuite aux membres du jury qui m'ont honoré par leur présence à ma soutenance de thèse. Je remercie tout particulièrement M. José RAGOT, Professeur à l'université de Lorraine, et M. Frédéric KRATZ, Professeur à l'INSA Centre Val de Loire, pour l'examen précis de ce manuscrit et pour leurs remarques enrichissantes qui m'ont permis d'éclaircir certains points difficiles et surtout d'ouvrir le champ des investigations futures.

Je remercie sincèrement Mme Ouiddad LABBANI-IGBIDA, Professeur à l'université de Limoges, d'avoir présidé le jury de ma soutenance de thèse, et Mme Agnès ROUSSY, maître de conférence à l'école des Mines Saint Etienn pour sa participation au jury en tant d'examinatrice.

Je souhaite remercier très vivement et sans exception, les membres du laboratoire LIS et en particulier les membres de mon équipe PECASE. Mes remerciements vont également aux doctorant(e)s du laboratoire pour leur gentillesse et sympathie.

Je ne saurais terminer sans remercier toute ma famille, plus particulièrement mes parents sans qui je n'aurais jamais su arriver jusque-là. Merci pour leur soutien et leur encouragement qui m'ont donné la force de continuer. Je remercie profondément tous ceux qui m'aiment et qui auraient voulu partager ma joie. Un grand merci à ma sœur et mon frère qui ont toujours été ma source de motivation afin de réaliser ma thèse de doctorat.

Enfin, je remercie du fond de mon cœur mon très cher mari de m'avoir toujours donné l'amour, le soutien et le courage durant les moments difficiles que j'ai rencontrés pendant cette thèse. Sans oublier mon très cher petit poussin Zakaria qui remplit ma vie de bonheur depuis son arrivée.

Cette thèse est le fruit de votre soutien et elle est pour vous !

TABLE DES MATIÈRES

Abstract	iii
Résumé	v
Travaux publiés	vii
Travaux soumis	ix
Remerciement	xi
Table des figures	xvii
Liste des tableaux	xxi
Nomenclature	xxiii
Introduction générale	1
I État de l'Art	9
I.1 Introduction	11
I.2 Classification des robots mobiles à roues	11
I.2.1 Types des roues	11
I.2.1.1 Les roues conventionnelles	11
I.2.1.2 Les roues omnidirectionnelles	12
I.2.2 Les robots non-holonômes et holonômes	13
I.2.2.1 Robots non-holonômes	13
I.2.2.2 Robots holonômes	15
I.3 Capteurs en robotique mobile	16
I.3.1 Les capteurs proprioceptifs	16
I.3.2 Les capteurs extéroceptifs	17
I.3.3 Les caméras	18
I.4 Terminologie et généralités	18
I.4.1 Terminologie sur le diagnostic	18
I.4.2 Généralités sur les <i>défauts</i> et le diagnostic	19

I.4.2.1	Classification des <i>défaux</i>	19
I.4.2.2	Classification des méthodes de diagnostic	22
I.4.2.3	Les résidus	25
I.4.2.4	Banc d'observateurs	27
I.5	Bibliographie sur le diagnostic des robots mobiles à roues	28
I.6	Bibliographie sur la navigation des robots mobiles à roues	30
I.7	Conclusion	33
II	Diagnostic des <i>défaux</i> des robots unicycles	35
II.1	Introduction	37
II.2	Modélisation des robots unicycles	37
II.2.1	Modèle cinématique et dynamique	39
II.2.1.1	Les contraintes non-holonômes	39
II.2.1.2	Modèle cinématique	41
II.2.1.3	Modèle dynamique	42
II.3	Commande des robots unicycles	43
II.4	Diagnostic des <i>défaux</i> capteurs et actionneurs des robots unicycles	44
II.4.1	Mesures disponibles et <i>défaux</i> étudiés	44
II.4.2	Système considéré	46
II.4.3	Principe de l'Observateur Etendu de Kalman (Extended Kalman Observer EKO)	48
II.4.4	Génération des résidus et table de signature	49
II.4.5	Résultats de simulation	53
II.5	Accommodation des <i>défaux</i> capteurs et actionneurs des robots unicycles	57
II.5.1	Résidus et table de signature	59
II.5.2	Estimation des <i>défaux</i> (Fault Estimation FE)	60
II.5.3	Compensation et accommodation des <i>défaux</i>	61
II.5.4	Résultats de simulation	62
II.6	Conclusion	68
III	Diagnostic des <i>défaux</i> des robots omnidirectionnels à 4 roues Mecanum	69
III.1	Introduction	71
III.2	Modélisation des robots omnidirectionnels (4-MWMR)	72
III.2.1	Modèle cinématique	72
III.2.2	Modèle dynamique	74
III.3	Commande des robots omnidirectionnels à 4 roues Mecanum (4-MWMRs)	75
III.4	Diagnostic des <i>défaux</i> capteurs et actionneurs des 4-MWMRs	78
III.4.1	Mesures disponibles et <i>Défaux</i> étudiés	79
III.4.2	Système augmenté	80
III.4.3	Génération des résidus et table de signature	82

III.4.4 Isolation des <i>défaux</i> actionneurs	84
III.4.5 Résultats de simulation	90
III.5 Isolation des <i>défaux</i> actionneurs des 4-MWMRs	92
III.5.1 Principe des observateurs à entrées inconnues (Unknown Input Observers ou UIOs) pour les systèmes linéaires à paramètres variants (LPVS)	93
III.5.2 Génération des résidus et table de signatures	97
III.5.3 Résultats de simulation	99
III.5.4 Sensibilité aux <i>défaux</i> capteurs	102
III.6 Surveillance de l'état de santé des actionneurs des 4-MWMRs et étude de son impact sur le comportement du robot	105
III.6.1 FDI des <i>défaux</i> actionneurs simultanés et identification des <i>défaux</i>	106
III.6.2 Simulation des <i>défaux</i>	107
III.6.3 Détermination des limites de sécurité	109
III.6.4 Résultats de simulation	110
III.6.4.1 Résultats de simulation du FDI des <i>défaux</i> actionneurs simultanés .	110
III.6.4.2 Résultats de simulation de la détermination des limites de sécurité .	111
III.6.5 Amélioration des limites de sécurité	114
III.6.6 Discussion	117
III.7 Conclusion	118
IV Reconfiguration de la trajectoire et compensation du retard généré par les obstacles mobiles	121
IV.1 Introduction	123
IV.2 Planification de missions et génération de trajectoire	124
IV.2.1 Planification des missions et du chemin (Missions and path planning)	124
IV.2.2 Génération de trajectoire	125
IV.3 Suivi de trajectoire et évitement d'obstacles	128
IV.3.1 Application aux robots omnidirectionnels à 4 roues Mecanum (4-MWMRs) .	128
IV.3.2 Évitement d'obstacles	129
IV.4 Reconfiguration de la trajectoire et gestion du retard	131
IV.4.1 Le principe de la gestion du retard	131
IV.4.2 Analyse de la compensation du retard	133
IV.5 Résultats de simulation	134
IV.5.1 Les scénarios	134
IV.5.2 Simulations	135
IV.5.3 Vérification et analyse de la possibilité de compensation du retard	137
IV.6 Conclusion	138
Conclusion	139

TABLE DES FIGURES

1	Wafer, (a) vierge (avant le <i>Front-End</i>), (b) gravé (après le <i>Front-End</i>)	1
2	Exemple d'un circuit intégré.	2
3	POD.	3
4	Quelques systèmes de transport automatiques utilisés dans la fab, source : (a) MURATEC, (b) IDC, (c) SCOTT, (d) MURATEC.	4
5	Les Work Packages du projet Productive4.0.	5
I.1	(a) Roue fixe, (b) Roue centrée orientable, (c) Roue décentrée orientable.	12
I.2	Roue omnidirectionnelle, a) Suédoise, b) Mecanum.	13
I.3	Présentation d'un robot mobile à roues dans son environnement.	13
I.4	Exemple de robot (a) tricycle, (b) type voiture.	14
I.5	Robot unicycle, OMRON LD90 et sa représentation dans un plan.	14
I.6	Robot omnidirectionnel, KUKA KMP, et sa représentation dans un plan.	15
I.7	Quelques mouvements réalisables par le robot KUKA KMP.	16
I.8	Différents niveaux d'apparition de <i>défauts</i>	19
I.9	Quelques <i>défauts</i> actionneur.	20
I.10	Quelques <i>défauts</i> capteur.	21
I.11	a) <i>Défaut</i> abrupt, b) <i>Défaut</i> graduel.	21
I.12	<i>Défaut</i> permanent et intermittent.	22
I.13	<i>Défaut</i> additif et multiplicatif.	22
I.14	Classification de quelques approches de diagnostic.	23
I.15	Principe d'utilisation de la redondance matérielle.	25
I.16	Structures <i>GOS</i> et <i>DOS</i> dans le cas des <i>défauts</i> actionneurs.	27
I.17	Structures <i>GOS</i> et <i>DOS</i> dans le cas des <i>défauts</i> capteurs.	28
II.1	Deux manières de représenter un robot mobile à roues dans son environnement.	38
II.2	Géométrie du robot unicycle.	40
II.3	Principe de détection et d'isolation des <i>défauts</i> étudiés.	51
II.4	Définition des seuils $\pm r_{th,1}$	54
II.5	Résidu r_1 en présence d'un <i>défaut</i> sur le codeur optique E_r	55
II.6	Résidus r_1, r_3, r_4 , et r_5 en présence d'un <i>défaut</i> sur l'actionneur W_r	55
II.7	Résidus r_6 et r_7 en présence d'un <i>défaut</i> gyroscope G_1	56

II.8 Résidus r_2, r_3, r_4, r_5, r_6 , et r_8 en présence d'un <i>défaut</i> actionneur W_l et d'un <i>défaut</i> gyroscope G_2	56
II.9 Résidus r_1, r_2, r_7 , et r_8 en présence de <i>défauts</i> codeurs droit et gauche, et d'un <i>défaut</i> gyroscope G_3	57
II.10 Classification des approches FTC.	58
II.11 Génération de trois résidus.	60
II.12 Principe du filtrage des résidus.	61
II.13 Principe de compensation des <i>défauts</i> capteurs et actionneurs des roues.	62
II.14 Détection et isolation d'un <i>défaut</i> au niveau du codeur de la roue droite avec le résidu r_1 avant (a) et après (b) filtrage.	63
II.15 Détection et isolation d'un <i>défaut</i> au niveau de l'actionneur de la roue gauche.	63
II.16 Estimation du <i>défaut</i> simulé sur W_l avec $r_{2,filtered}$	64
II.17 Compensation d'un <i>défaut</i> simulé au niveau du codeur E_l	64
II.18 Chemin de déplacement du robot unicycle selon l'axe x et y comparés aux chemins désirés, en absence de <i>défauts</i>	65
II.19 Chemin de déplacement du robot unicycle $y(x)$ comparé au chemin désiré, en absence de <i>défauts</i>	65
II.20 Signal de commande u	66
II.21 Variation de l'angle d'orientation du robot θ	66
II.22 Chemin du robot en présence d'un <i>défaut</i> au niveau de l'actionneur de la roue droite W_r , sans et avec compensation du <i>défaut</i> , comparé au chemin désiré.	67
II.23 Chemin du robot en présence d'un <i>défaut</i> abrupt au niveau de l'actionneur de la roue gauche W_l , sans et avec compensation du <i>défaut</i> , comparé au chemin désiré.	67
III.1 Géométrie du robot omnidirectionnel.	72
III.2 Suivi du profile de trajectoire $y(x)$ en utilisant la commande par retour d'état linéarisant.	77
III.3 Suivi de trajectoire $T_{ref}(t)$ en utilisant la théorie de commande par retour d'état linéarisant.	77
III.4 Évolution du contrôle appliqué au robot.	78
III.5 Variation de l'angle du robot θ	78
III.6 Génération des résidus pour la détection et l'isolation des <i>défauts</i> actionneurs et capteurs du 4-MWMR.	84
III.7 Signature des 5 résidus en présence d'un <i>défaut</i> au niveau du codeur E_2	90
III.8 Résidu r_1 en présence d'un <i>défaut</i> gyroscope G	91
III.9 Résidus r_2 et r_5 en présence d'un <i>défaut</i> au niveau de l'actionneur A_1	91
III.10 Résidus r_3 et r_4 en présence d'un <i>défaut</i> au niveau de l'actionneur A_3	92
III.11 Illustration de la convergence du UIO vers la valeur réelle et son erreur d'estimation.	97
III.12 Banc d'observateurs UIOs sous la structure GOS pour le FDI des <i>défauts</i> actionneurs.	98

III.13	Les résidus en mode nominal, absence de <i>défaux</i>	99
III.14	Le résidu r_1 en présence d'un <i>défaut</i> actionneur A_1	100
III.15	Le résidu r_2 en présence d'un <i>défaut</i> actionneur A_2	100
III.16	Le résidu r_3 en présence d'un <i>défaut</i> actionneur A_3	101
III.17	Le résidu r_4 en présence d'un <i>défaut</i> actionneur A_4	101
III.18	Les quatre résidus en présence de <i>défaux</i> capteurs.	103
III.19	Suivi de trajectoire en présence des <i>défaux</i> capteurs.	104
III.20	Estimation de la perte d'efficacité de l'actionneur A_1	107
III.21	Simulation des dégradation des configurations.	108
III.22	Suivi de trajectoire sous différentes conditions de fonctionnement.	110
III.23	Sauvegarde des erreurs maximales de toutes les configurations dans un vecteur.	111
III.24	Détermination des limites des dégradations acceptées.	112
III.25	Les résidus en présence d'un <i>défaut</i> au niveau de l'actionneur de la roue 2 et 3.	112
III.26	Les résidus en présence d'un <i>défaut</i> au niveau de l'actionneur de la roue 1, 2 et 4.	113
III.27	Les résidus en présence d'un <i>défaut</i> au niveau des actionneurs des quatre roues.	113
III.28	Limites des dégradations acceptées dans le cas avec quatre actionneurs en absence et en présence d'obstacles.	114
III.29	Limites des dégradations acceptées dans le cas avec trois actionneurs en absence et en présence d'obstacles.	115
III.30	Scenario #3 comparé au Scenario #1 en absence d'obstacles.	116
III.31	Limites de dégradation obtenues pour Scenario #3 pour les deux cas d'études: avec quatre et avec trois actionneurs.	116
III.32	Limites de dégradation obtenues pour Scenario #2 en évitant les obstacles avec $V = 1$ m/s pour les deux cas d'études: avec quatre et avec trois actionneurs.	117
IV.1	Principe de calcul de $t_{i,min}$ pour la vérification de la faisabilité d'une sous-mission m_i	125
IV.2	Principe de génération des trajectoires de vitesse et de position.	127
IV.3	Suivi de trajectoire en utilisant la méthode de retour d'état linéarisant.	128
IV.4	Suivi de $x_{ref}(t)$ et de $y_{ref}(t)$ en utilisant la méthode de retour d'état linéarisant.	129
IV.5	Principe de la stratégie d'évitement d'obstacles.	130
IV.6	Compensation du retard (Exemple 1).	132
IV.7	Compensation du retard (Exemple 2).	133
IV.8	Schéma fonctionnel du principe de la compensation du retard.	134
IV.9	Trajectoire de la mission globale sans et avec les obstacles.	135
IV.10	Trajectoires de vitesse et de position pour tous les scénarios étudiés.	136
IV.11	Trajectoires $x_{ref}(t)$ et $y_{ref}(t)$ des quatre scénarios étudiés.	136
IV.12	Estimation du retard pouvant être compensé à chaque instant pour les quatre scénarios étudiés.	137

LISTE DES TABLEAUX

3	Les tasks du WP8	6
I.1	Principe d'isolation des <i>défauts</i> capteur en utilisant la redondance matérielle.	25
II.1	Définition des variables du modèle mathématique du robot unicycle.	40
II.2	Table de signature des résidus décisionnels en présence des différents <i>défauts</i> capteurs et actionneurs étudiés.	52
II.3	Table de signature des résidus décisionnels en présence de quelques <i>défauts</i> simultanés. .	53
II.4	La signature des trois résidus en présence des <i>défauts</i> codeurs et actionneurs.	60
III.1	Définition des variables du modèle mathématique du robot omnidirectionnel.	73
III.2	Le principe de quelques mouvements des robots 4-MWMRs.	79
III.3	Signature des cinq résidus en présence de <i>défauts</i> actionneurs et capteurs.	84
III.4	Les paramètres physiques de la plateforme KUKA-KMR.	88
III.5	Pourcentage de détection des <i>défauts</i> actionneurs par chaque résidus r_i , ($i = 2, \dots, 5$). .	89
III.6	Pourcentage de détection des <i>défauts</i> actionneurs par chaque résidus r_i , ($i = 2, \dots, 5$). .	89
III.7	Signature des résidus décisionnels en présence des <i>défauts</i> actionneurs.	98
III.8	Quelques configurations de dégradations pour le même niveau de dégradation.	117

NOMENCLATURE

Abréviation

fab Les installations de fabrication du semi-conducteur, appelées aussi "fabrication facility"

FOUPs Front Opening Unified Pods, souvent appelés Pods

KO Kalman Observer

UIO Unknown Input Observer

WMRs Robots mobiles à roues, ou Wheeled mobile robots

INTRODUCTION GÉNÉRALE

Contexte industriel

Ce travail de recherche s'inscrit dans le cadre du projet Européen *Productive4.0*. Il a été effectué au sein du laboratoire *LIS UMR 7020* (Laboratoire d'Informatique et Systèmes) à Polytech Marseille dans l'équipe PECASE (Pronostic/diagnostic Et CommAnde : Santé et Energie) en collaboration avec l'entreprise *STMicroelectronics* du site de Rousset, un des leaders mondiaux de la fabrication du semi-conducteur.

Les dispositifs semi-conducteurs sont omniprésents dans notre vie quotidienne. Ainsi, on les trouve dans les ordinateurs, les TVs, les voitures, les smart-phones, etc. Ces dispositifs sont composés principalement de deux parties. La première partie est appelée **la puce** (ou *chips*). C'est un carré de silicium caractérisé par sa fragilité et sa faible épaisseur. Quant à la deuxième partie, elle correspond à un boîtier (ou *package*) qui protège la puce de son environnement externe et qui facilite son montage dans les systèmes.

Étapes de fabrication des circuits intégrés : La matière de base d'un dispositif semi-conducteur est généralement des plaquettes ou galettes de silicium, nommées *wafers* (voir Figure 1 (a)), et qui servent de support à la fabrication des puces (d'autres matériaux semi-conducteurs peuvent également être utilisés tel que le germanium ou l'arsénure de gallium).

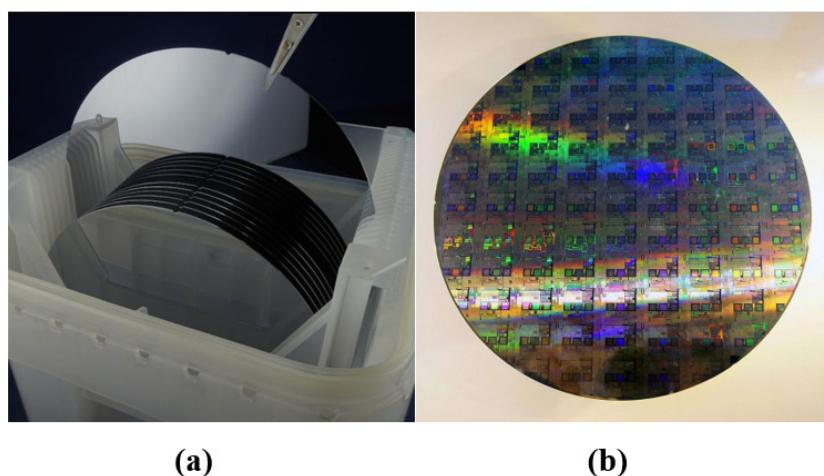


FIGURE 1 – Wafer, (a) vierge (avant le *Front-End*), (b) gravé (après le *Front-End*).

La fabrication des composants micro-électroniques passe par deux grandes étapes importantes :

a) Le *Front-End*, qui regroupe les étapes de fabrication des éléments de base de la puce. Ces étapes peuvent varier en fonction de la technologie et du type de circuit. Néanmoins, il existe quelques étapes élémentaires du processus communes à la plupart des technologies (e.g. oxydation, dépôt de résine, photolithographie, développement, gravure, dopage, ...). A noter que selon le produit, ces étapes peuvent se répéter plusieurs fois. Le résultat final du *Front-End* est un wafer sur lequel sont implémentées les puces (voir Figure 1 (b)).

b) Le *Back-End*, qui regroupe les étapes d'encapsulage des puces dans un boîtier approprié afin de produire un circuit prêt à la vente (voir Figure 2). Les différentes étapes du *Back-End* sont : la découpe, le collage, le câblage, l'encapsulage et la mise en forme, enfin le marquage et le test.



FIGURE 2 – Exemple d'un circuit intégré.

D'une manière générale, la fabrication des puces requiert environ 600 étapes de fabrication¹ qui ne se déroulent pas forcément toutes au même endroit. Par exemple, après une isolation dans la zone de photolithographie, les lots subissent généralement une inspection dans une autre zone, celle de métrologie [10]. La tâche qui correspond au déplacement des lots d'une zone de fabrication à une autre est appelée "transport".

Transport des wafers : Le transport des wafers s'effectue par lots dans des boîtes hermétiques dites **PODs** (voir Figure 3), chaque POD comprend au maximum 25 wafers de diamètre de 200mm. Avec le temps, et afin d'augmenter le rendement de la production, les fabricants du semi-conducteur utilisent des wafers de plus en plus grands 300mm afin de pouvoir y graver davantage de puces simultanément et de limiter les pertes sur le bord des plaquettes. Dans ce cas, le transport des wafers s'effectue dans ce qu'on appelle les **FOUPs**², chaque FOUP comprend au maximum 25 wafers de diamètre de 300mm. Ainsi, la technologie du semi-conducteur progresse avec l'évolution du diamètres des wafers.

Avant, le transport des lots entre les équipements de fabrication était assuré par les opérateurs humains. Avec l'augmentation du diamètres des wafers, les lots deviennent plus lourds à transporter. Par exemple, avec la technologie de 300 mm, un FOUP plein pèse environ 15 kg, contre 5 kg

1. Une étape de fabrication correspond au passage d'un lot de wafers dans une machine

2. Front Opening Unified Pods



FIGURE 3 – POD.

seulement avec l'ancienne technologie (200 mm) [10]. Cela rend la tâche répétitive du transport plus fastidieuse, et augmente les risques de troubles musculo-squelettiques³ chez les opérateurs.

Les systèmes de transport automatiques représentent une excellente alternative à l'humain. D'une part parce qu'ils peuvent transporter des charges plus importantes, et d'une autre part parce qu'ils sont rapides, précis, et ils peuvent fonctionner sur de longues durées sans s'arrêter. Pour ces raisons, l'industrie du semi-conducteur aujourd'hui utilise de plus en plus des systèmes automatiques afin de transporter les PODs ou les FOUPs entre les équipements et les processus de fabrication.

Les systèmes de transport automatiques ont tous la même utilité dans la fab⁴ : transporter des FOUPs d'un point à un autre. Mais en fonction de la situation, certains types de véhicules sont mieux adaptés que d'autres. Parmi les systèmes de transport automatiques utilisés en fab, on peut citer :

- a) les véhicules sur rail (ou Rail Guided Vehicles) : Ce sont des véhicules qui se déplacent souvent sur les rails et qui sont indépendants de l'humain. Leur particularité est qu'ils sont utilisés pour de courts trajets (entre deux équipements en général), voir Figure 4 (a).
- b) les véhicules guidés par l'humain (ou Person Guided Vehicles) : Ce sont des véhicules complètement dirigés par l'humain. Ils sont utilisés par les opérateurs pour transporter les FOUPs aux endroits où il n'y a pas de système automatique, voir Figure 4 (b).
- c) les véhicules guidés automatiquement (ou Automated Guided Vehicles) : Ce sont des véhicules complètement indépendants de l'humain, voir Figure 4 (c).

3. Maladies touchant les articulations, les muscles et les tendons. Effectuer des tâches répétitives est l'un des facteurs déclencheurs de ces maladies

4. L'abréviation 'fab' est utilisée afin de désigner les installations de fabrication du semi-conducteur, appelées aussi "fabrication facility"

- d) les véhicules de transport aérien (ou OverHead Transport Vehicles) : Ce sont des véhicules qui se déplacent au plafond. Cela permet de gagner de la place au sol et d'avoir plus d'équipements de production à l'intérieur de l'unité de fabrication, voir Figure 4 (d).

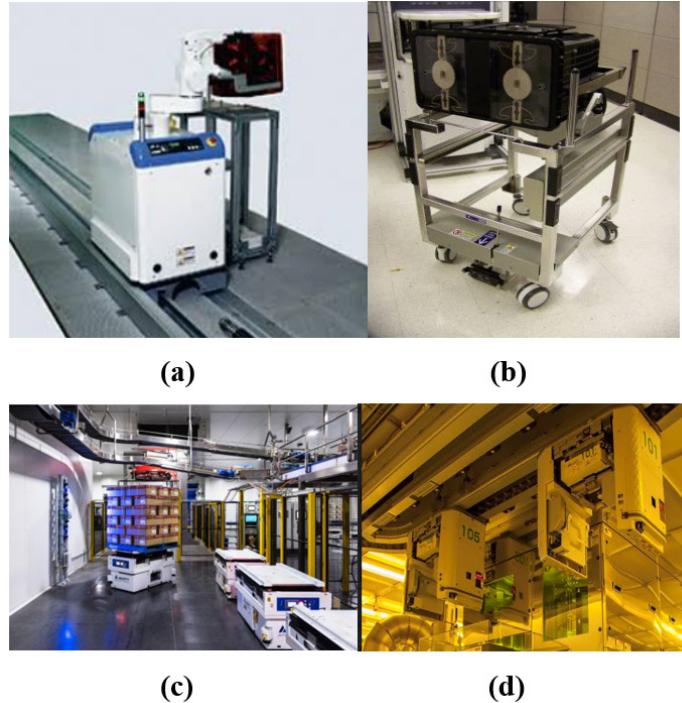


FIGURE 4 – Quelques systèmes de transport automatiques utilisés dans la fab, source : (a) MURATEC, (b) IDC, (c) SCOTT, (d) MURATEC.

Dans le cadre du projet Européen *Productive4.0*, *STMicroelectronics* de Rousset a décidé d'utiliser des robots mobiles à roues dans les endroits où les systèmes automatiques cités précédemment ne peuvent pas être intégrés, afin que le transport des PODs soit complètement automatisé dans la fab.

Le projet Européen *Productive4.0* a pour objectif l'amélioration de l'industrie 4.0. Il met en collaboration plus de 100 partenaires académiques et industriels dans 19 pays pour un budget dépassant 100 millions d'euros sur trois ans. Le projet contient 10 lots de travaux ou Work Packages notés WPs (voir Figure 5). Chaque WP a un ensemble d'objectifs techniques (Tasks) à atteindre d'une manière itérative et progressive tout au long de la période du projet.

STMicroelectronics de Rousset et le *LIS UMR 2070* ont proposé une collaboration permettant de moderniser la fab au travers de sa robotisation et d'apporter des aspects de sécurité via la surveillance de l'état de santé des robots dans le cadre du WP8 qui est un WP qui contient 25 tasks (voir Table 3). Les travaux de cette thèse s'inscrivent plus précisément dans la Section T8.3.4 dénommée *Robotisation de la fab*, ou *Fab Robotization*, dont les objectifs portent en particulier sur :

- Le contrôle de l'état de santé des robots intégrés dans une fab,

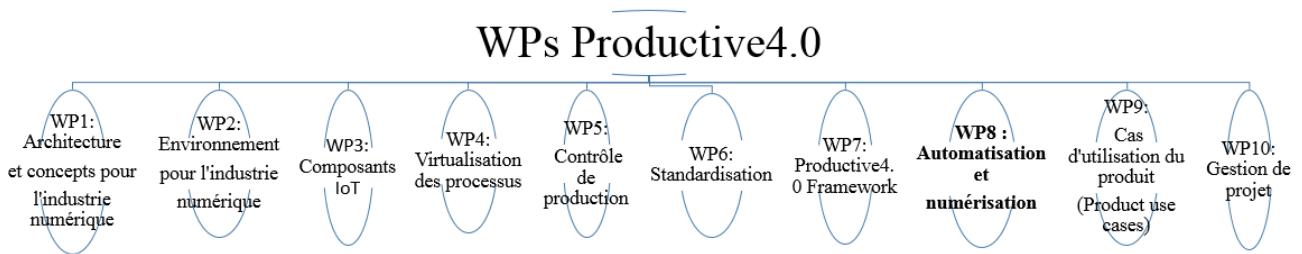


FIGURE 5 – Les Work Packages du projet Productive4.0.

- La reconfiguration des robots,
- Le contrôle et l'étude des performances d'une flotte de robots dans une fab,
- La reconfiguration et la gestion de la flotte de robots.

Choix des robots Après avoir fait des benchmarks chez des partenaires et des concurrents, et après étude du marché, *STMicroelectronics* a abouti au choix de deux types de robots, notamment un robot unicycle (non-holonôme) de la société OMRON-Adept, et un robot omnidirectionnel à 4 roues Mecanum (robot holonôme) de la société KUKA. La différence entre ces deux types de robots (non-holonôme et holonôme) sera détaillée dans le **Chapitre I**.

Problématique

Le but de la robotisation de la fab est liée à l'optimisation de la productivité et à l'amélioration du confort de la production. En effet, comparés aux opérateurs humains, les robots sont plus précis, plus rapides, et surtout ils peuvent fonctionner sur de longues durées sans s'arrêter ou être fatigués. Néanmoins, malgré tous les avantages que présentent ces robots, leur utilisation dans la fab est à double tranchant. En effet, afin d'être efficaces, ces véhicules doivent être capables de naviguer et de réaliser leurs tâches d'une façon autonome dans la fab (*i.e.*, sans aucune assistance ou intervention humaine). De plus, il existe divers défis à relever, particulièrement :

1. Robots et interaction humaine : L'objectif de ce défi est d'étudier les interactions entre robots et entre robots et humains (opérateurs de la fab) ainsi que d'apporter des outils de sécurité. De plus, les collaborations entre robots ou robots et humains doivent être prises en compte pour réussir une mission.
2. Surveillance de l'état de santé des robots : L'objectif de ce défi est de répondre à la question «Comment être sûr que le robot réussit sa mission?». Quand un *défaut* apparaît, il peut impacter considérablement le comportement du robot et l'empêcher de réaliser sa mission. De plus, cela peut présenter un réel danger, que ce soit pour le robot lui-même ou pour les équipements et

TABLE 3 – Les tasks du WP8

Task N°	Description
T8.1.1	Intégration de l'individualisation des véhicules dans un processus d'assemblage hautement automatisé dans l'industrie automobile dans la logistique basée sur le cycle de vie des produits
T8.1.2	Robots aériens
T8.1.3	Système IoT
T8.1.4	Services de suivi, de détection et d'actionnement
T8.2.1	Services intelligents pour les équipements de test
T8.2.2	Ingénierie des coûts simultanés pour les architectures de groupe motopropulseur
T8.2.3	Services intelligents pour un site de fabrication fiable
T8.2.4	Gestion de la chaîne d'approvisionnement pour la fabrication de semi-conducteurs
T8.2.5	Laboratoire intelligent d'analyse des pannes
T8.3.1	Technologie de refroidissement double face frontale / backend
T8.3.2	Solution d'automatisation élevée dans la ligne de production de plaquettes SC
T8.3.3	Analyse de données, manque de données Semi-conducteur
T8.3.4	Robotisation de la fab
T8.3.5	Supervision en usine pour la réduction de la variabilité
T8.3.6	Systèmes robotiques mobiles adaptatifs pour une fabrication intelligente
T8.3.7	Automatisation intelligente de la production de semi-conducteurs par des robots autonomes flexibles avec une fonctionnalité de gestion avancée
T8.3.8	Manipulation adaptative et intelligente des substrats
T8.3.9	Cadre d'automatisation pour les nouveaux équipements sans normes d'automatisation
T8.3.10	Cadre d'automatisation et stratégies d'automatisation pour les procédures avancées de nettoyage des porteurs pour les substrats semi-conducteurs
T8.3.11	Optimisations de la production mondiale et locale en temps réel
T8.3.12	Réseau d'optimisation utilisant la planification avancée
T8.3.13	Méthodes modernes de détection des valeurs aberrantes pour la fabrication de semi-conducteurs
T8.3.14	Gestion du contrôle, des tâches et des décisions automatisés de fab
T8.3.15	L'humain dans une zone de fabrication numérisée 4.0
T8.3.16	Suivi d'un seul appareil et contrôle de processus avancé dans l'assemblage et l'emballage pour l'intégration du système

les opérateurs de la fab. Par conséquence, il est important d'avoir une connaissance de l'état de santé du robot, et ceci à travers le développement des indicateurs de santé du robot en fonction de l'occurrence de défauts impactant son bon fonctionnement. Ces indicateurs permettront de prendre des décisions si le robot peut ou non réaliser sa mission, et de développer des outils afin de compenser l'impact des défauts sur le comportement du robot. Le contrôle de la santé du robot a un impact important sur la performance de la fab.

3. Gestion du retard généré par les obstacles imprévus sur le chemin du robot : la fab est un environnement dynamique. Elle contient des obstacles fixes, mais aussi des obstacles mobiles qui ne peuvent être prévus sur le chemin du robot durant la navigation. Ainsi, ces derniers peuvent perturber les missions du robot en générant un retard par rapport à la durée initiale

planifiée de la mission. Cela peut avoir comme conséquence d'impacter significativement le rendement de la chaîne de production.

4. Reconfiguration de la flotte de robots : lorsqu'un problème survient sur un robot, cela le rend temporairement indisponible pour réaliser de sa mission. Une reconfiguration de stratégie de la flotte est nécessaire. Des décisions doivent être prises sur la manière de reconfigurer l'activité de la flotte de robots afin d'obtenir toujours une performance optimale.

Cette thèse portera essentiellement sur les deux points 2 et 3 « surveillance de l'état de santé des robots et gestion du retard généré par les obstacles imprévus sur le chemin du robot».

Contributions apportées

Dans cette thèse, nous nous intéressons dans un premier lieu aux robots unicycles et au diagnostic et à la compensation des *défaux* actionneurs et capteurs à la fois. Le challenge par rapport à la littérature est de distinguer les *défaux* actionneurs des *défaux* capteurs dont l'information de l'état de fonctionnement est obtenue via la même source d'information (capteurs de vitesse angulaire des roues).

Dans un second temps, nous nous focalisons sur les robots omnidirectionnels, et nous nous intéressons au diagnostic des *défaux* capteurs et actionneurs et à l'étude de l'impact des *défaux* sur le comportement des robots. Le diagnostic de ce type de robots est un sujet qui n'a pas été largement traité dans la littérature car les robots holonômes (omnidirectionnels) sont assez récents comparés aux robots non-holonômes.

Enfin, nous proposons une stratégie permettant de compenser/réduire le retard généré par les obstacles imprévus sur le chemin des robots, avec une application numérique sur les robots omnidirectionnels. La compensation du retard des systèmes de transport dans l'industrie est une problématique qui n'est pas traitée dans la littérature. À noter que la méthodologie développée dans ce travail de thèse reste efficace et adaptable à une large sélection de véhicules autonomes tels que les robots unicycles car elle est indépendante du modèle mathématique du robot.

Organisation du manuscrit

Ce manuscrit est organisé en quatre chapitres :

Le **Chapitre I** commence par un état de l'art des différents types de robots mobiles à roues (Section I.2), suivi d'un listing des capteurs les plus couramment utilisés dans la robotique mobile (Section I.3). Nous présentons ensuite des généralités sur les *défaux* et les méthodes de diagnostic (Section I.4), suivi d'un état de l'art des différents travaux proposés par les chercheurs au cours du temps sur la thématique du diagnostic des *défaux* des robots mobiles à roues (Section I.5). Enfin, nous introduisons dans la Section I.6 la deuxième problématique portant sur la compensation du

retard durant la navigation en présentant une bibliographie sur les méthodes de navigation des robots mobiles à roues (*i.e.* planification de chemin, évitement d'obstacles, ...).

Le **Chapitre II** est consacré aux travaux effectués sur les robots unicycles dans le but de répondre à la problématique du diagnostic (*i.e.* détection, isolation, et identification) et de l'accommodation à la fois des *défauts* capteurs et des *défauts* actionneurs [6, 4]. Des résultats de simulation sont donnés après la description de chaque approche afin d'illustrer son efficacité.

Le **Chapitre III** est dédié aux travaux effectués sur les robots omnidirectionnels à 4 roues Mecanum afin de traiter la problématique du diagnostic des *défauts* [5, 2, 7, 8]. Le travail présenté dans [5] considère à la fois les *défauts* capteurs et les *défauts* actionneurs. A l'issu de l'approche proposée dans ce travail, les *défauts* capteurs sont facilement isolables, tandis que les *défauts* actionneurs prennent plus du temps pour être isolés. Ensuite, nous nous intéressons aux *défauts* actionneurs et nous proposons une seconde approche dans [2] afin de faciliter leur isolation. Après cela, nous nous focalisons particulièrement sur la dégradation des actionneurs et nous proposons dans [7, 8] une méthodologie permettant de surveiller l'évolution de l'état de santé des actionneurs après une première détection de *défauts*, puis d'étudier son impact sur le comportement du robot, dans le but de définir des limites de sécurité liées au niveau de dégradation que l'on peut avoir au niveau des actionneurs, sans que cela impacte le comportement du robot. Ces trois travaux sont détaillés dans ce chapitre, et chaque approche proposée est suivie de résultats de simulation.

Le **Chapitre IV** détaille l'approche que nous avons proposée afin de réduire ou de compenser complètement (quand c'est possible) le retard généré par la rencontre des obstacles mobiles et imprévus sur le chemin des robots durant la navigation [3, 1, 9]. Cette approche est basée sur la reconfiguration de la trajectoire en vitesse. Cette dernière se fait en temps réel, et s'adapte aux changements des conditions de fonctionnement et de l'environnement.

Ce manuscrit s'achève sur une conclusion générale, qui dresse une synthèse des travaux réalisés, des verrous levés, ainsi que des résultats obtenus. De plus, les contributions principales décrites dans chaque chapitre sont mises en évidence, autant du point de vue de la recherche que de celui du développement et de l'adaptation au contexte du projet *Productive4.0* et de *STMicroelectronics*. Enfin, une ouverture vers des perspectives de recherches sont données, elle concerne l'utilisation des indices de dérives ou des intervalles de sécurité (obtenus de l'étude de l'évolution de l'état de santé des actionneurs et de l'analyse de son impact sur le comportement des robots) pour le pronostic de défaillances, ainsi que la généralisation de l'approche proposée pour la gestion du retard sur le cas d'un ensemble de robots mobiles fonctionnant en collaboration pour répondre aux besoins de l'entreprise.

CHAPITRE I

ÉTAT DE L'ART

I.1	Introduction	11
I.2	Classification des robots mobiles à roues	11
I.2.1	Types des roues	11
I.2.1.1	Les roues conventionnelles	11
I.2.1.2	Les roues omnidirectionnelles	12
I.2.2	Les robots non-holonômes et holonômes	13
I.2.2.1	Robots non-holonômes	13
I.2.2.2	Robots holonômes	15
I.3	Capteurs en robotique mobile	16
I.3.1	Les capteurs proprioceptifs	16
I.3.2	Les capteurs extéroceptifs	17
I.3.3	Les caméras	18
I.4	Terminologie et généralités	18
I.4.1	Terminologie sur le diagnostic	18
I.4.2	Généralités sur les <i>défauts</i> et le diagnostic	19
I.4.2.1	Classification des <i>défauts</i>	19
I.4.2.2	Classification des méthodes de diagnostic	22
I.4.2.3	Les résidus	25
I.4.2.4	Banc d'observateurs	27
I.5	Bibliographie sur le diagnostic des robots mobiles à roues	28
I.6	Bibliographie sur la navigation des robots mobiles à roues	30
I.7	Conclusion	33

I.1 Introduction

Les robots mobiles à roues sont de plus en plus sollicités dans l'industrie, notamment pour le transport d'objets ou de produits. Ils sont utilisés non seulement pour leur rapidité et leur précision, mais aussi pour leur capacité à travailler pendant plusieurs heures sans s'arrêter. De plus, ils peuvent supporter des charges très importantes comparés à l'Homme.

Compte tenu du nombre d'actionneurs qu'un robot possède ainsi que le type de ses roues, il peut être "**non-holonôme**" ou "**holonôme**". La définition et la comparaison de ces deux catégories des robots mobiles à roues fera l'objet de la première partie de ce chapitre (Section I.2), où des exemples de robots pour chaque catégorie seront donnés. Avant de détailler ces deux catégories, nous donnons une description des différents types des roues utilisées dans la robotique mobile à roues. La section suivante (Section I.3) sera consacrée à la définition d'un ensemble de capteurs souvent utilisés dans la robotique mobile, suivi de généralités sur les *défauts*, les méthodes du diagnostic, et les propriétés des résidus dans la Section I.4 .

Enfin, un état de l'art des travaux de la littérature traitant les deux problématiques étudiées dans ce manuscrit, *i.e.* 1) Le diagnostic des *défauts*, et 2) Gestion du retard généré par les obstacles mobiles, sera présenté. Concernant la première problématique, différentes références bibliographiques sur le diagnostic des *défauts* des robots mobiles à roues seront définies dans la Section I.5, pendant qu'une bibliographie sur les méthodes de navigation, comprenant la planification de chemin, de trajectoire, et l'évitement d'obstacles sera fournie dans la Section I.6.

I.2 Classification des robots mobiles à roues

Les robots mobiles à roues sont généralement divisés en deux classes principales, 1) Les robots dits **non-holonômes**, et 2) Les robots dits **holonômes**. Dire qu'un robot appartient à la première ou à la deuxième classe dépend du type de ses roues ainsi que du nombre de ses degrés de liberté (Degrees of freedom, DOFs) contrôlables comparé au nombre total de ses DOFs, qui définit sa mobilité dans son espace d'évolution. D'une façon générale, nous rappelons qu'un robot évoluant sur une surface plane a 3 DOFs, deux translations et une rotation [11].

I.2.1 Types des roues

Le mode de locomotion des robots mobiles à roues se définit par le choix et la combinaison de leurs roues. On rencontre principalement deux classes différentes : Les roues dites "conventionnelles", et les roues dites "omnidirectionnelles" [12].

I.2.1.1 Les roues conventionnelles

Trois types de roues conventionnelles sont distingués [12, 13] (voir Figure I.1).

1. *Les roues fixes*, dont l'axe de rotation est fixe et passe par le centre de la roue. Ces roues peuvent tourner uniquement autour de leur axe principal de rotation, voir Figure I.1(a).

2. *Les roues centrées orientables*, dont l'axe d'orientation n'est pas fixe et passe par le centre de la roue. Ces roues peuvent tourner autour de leur axe principal de rotation et autour du point de contact noté P , voir Figure I.1 (b).

3. *Les roues décentrées orientables*, souvent appelées *roues folles*, pour lesquelles l'axe d'orientation ne passe pas par le centre de la roue. Ces roues peuvent tourner autour de leur axe principal de rotation, autour du point de contact P , et aussi autour du pivot, voir Figure I.1 (c).

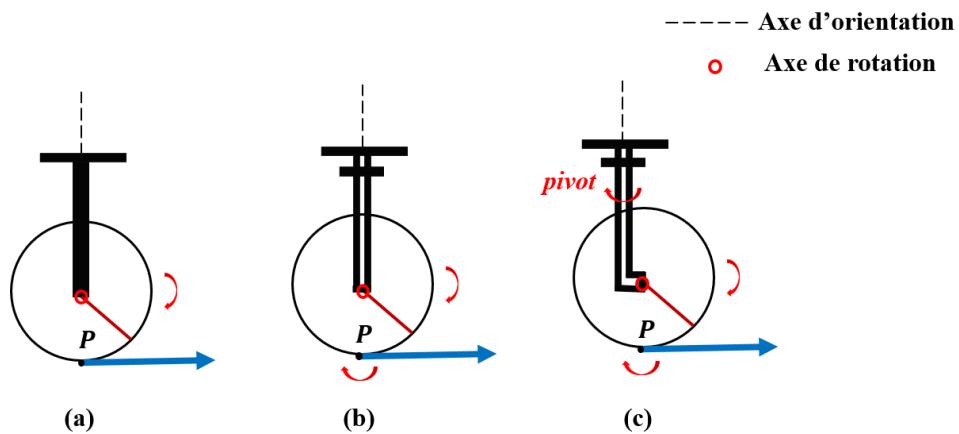


FIGURE I.1 – (a) Roue fixe, (b) Roue centrée orientable, (c) Roue décentrée orientable.

I.2.1.2 Les roues omnidirectionnelles

Les roues omnidirectionnelles sont distinguées des roues conventionnelles par leur structure assez particulière. Elles sont composées d'une roue centrale et de rouleaux libres (Free rollers) montés tout autour de la périphérie de cette dernière. Elles ont été inventées en 1975 par Ilon, un ingénieur Suédois [14]. Selon le type des rouleaux ainsi que leur angle d'inclinaison par rapport au périmètre de la roue principale, plusieurs designs des roues omnidirectionnelles peuvent-être différenciés [15–18].

Parmi ces roues, deux types sont fréquemment utilisés sur des plateformes mobiles. Les roues dites "Suédoises" (ou Swedish wheels) [19], et les roues dites "Mecanum" (ou Mecanum wheels). Elles sont caractérisées par des rouleaux sphériques et un angle d'inclinaison $\gamma = 0^\circ$ et $\gamma = 45^\circ$ respectivement, voir Figure I.2.

Les roues Suédoises sont généralement utilisées pour les petites plateformes mobiles telle que les robots de football (Soccer robots) [20], tandis que les roues Mecanum sont plus appropriées pour des plateformes utilisées dans le transport d'objets lourds dans l'industrie [21].

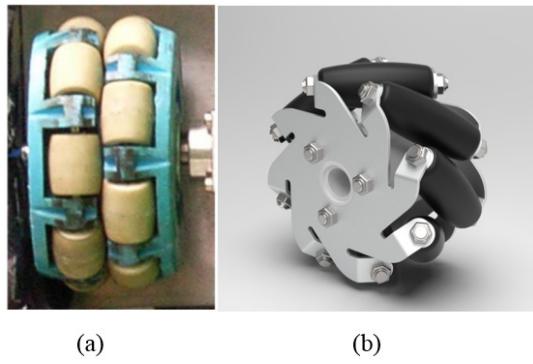


FIGURE I.2 – Roue omnidirectionnelle, a) Suédoise, b) Mecanum.

I.2.2 Les robots non-holonômes et holonômes

Afin d'expliquer la différence entre ces deux classes de robots mobiles à roues, on considère le robot schématisé par un rectangle dans la Figure I.3, dont on suppose que tous les mouvements sont contenus dans un plan de navigation parallèle au sol. Le repère absolu $(O, \vec{x}, \vec{y}, \vec{z})$, avec \vec{x} l'axe des abscisses, \vec{y} l'axe des ordonnées, et \vec{z} l'axe vertical, est un repère fixe et lié à l'environnement du robot. Il est appelé repère global ou de référence, tandis que $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ est un repère local, lié au centre de masse du robot G . C'est un repère mobile et son déplacement est lié à celui du point G .

Le vecteur $[x, y, \theta]^T$ est appelé "posture du robot" [22] dans le repère global $(O, \vec{x}, \vec{y}, \vec{z})$, où x et y représentent respectivement l'abscisse et l'ordonnée du point G (*i.e.* la position du robot suivant l'axe \vec{x} et \vec{y} respectivement), tandis que θ représente l'angle que forme le robot avec l'axe \vec{x} (*i.e.* l'orientation du robot).

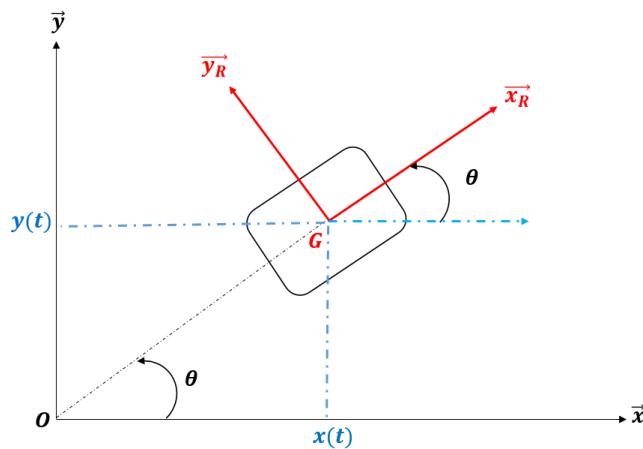


FIGURE I.3 – Présentation d'un robot mobile à roues dans son environnement.

I.2.2.1 Robots non-holonômes

Cette catégorie de robots est appelée ainsi du fait des contraintes liées au roulement sans glissement [23, 24] qui seront détaillées dans la Section II.2.1.2. Ces contraintes ne peuvent pas être intégrées,

d'où leur nom : contraintes non intégrables ou non-holonômes. De manière concrète, l'existence de contraintes non holonômes implique que le système ne peut pas effectuer certains mouvements instantanément. C'est pourquoi, les robots non-holonômes sont limités à 2 DOFs contrôlables dans un plan : translation suivant l'axe \vec{x} et rotation autour de l'axe \vec{z} [11], voir Figure I.3. Ils sont généralement composés d'un certain nombre de roues commandées et plusieurs roues folles pour maintenir leur stabilité.

Parmi les robots non-holonômes, on peut citer les robots tricycles avec trois roues actionnées et les robots type voiture avec quatre roues motrices (voir Figure I.4), ainsi que les robots unicycles qui nous intéressent plus particulièrement dans ce manuscrit. Les contraintes non-holonômes se traduisent dans l'incapacité de ces robots de se déplacer parallèlement à l'axe de la roue (*i.e.* l'axe \vec{y}_R) d'une façon instantanée. Un tel déplacement nécessitera des manœuvres. De même, pour les robots type voiture par exemple, ils ne peuvent pas se garer sans effectuer des crêneaux.



FIGURE I.4 – Exemple de robot (a) tricycle, (b) type voiture.

Dans les travaux de cette thèse, nous nous intéressons aux robots unicycles, qui sont généralement actionnés par deux roues indépendantes et qui possèdent éventuellement un certain nombre de roues folles assurant sa stabilité. Plus précisément, nous étudions le robot OMRON LD90 de la société OMRON-Adept, qui est une plateforme avec deux roues motrices fixes placées au milieu du châssis pour que le robot puisse tourner sur lui-même, et deux roues folles placées à l'avant et à l'arrière du robot pour assurer sa stabilité comme montré sur la Figure I.5.

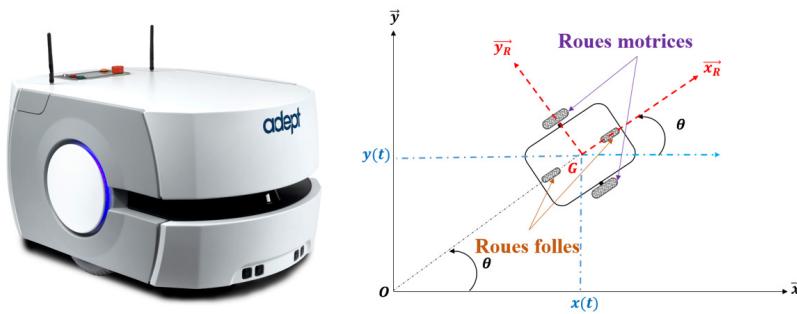


FIGURE I.5 – Robot unicycle, OMRON LD90 et sa représentation dans un plan.

I.2.2.2 Robots holonômes

Le nombre de DOFs contrôlables d'un robot holonôme est égal au nombre total des DOFs qui peuvent exister dans un plan (*i.e.* 3 DOFs) : en plus de la translation suivant l'axe \vec{x} et de la rotation suivant l'axe \vec{z} , les robots holonômes peuvent se déplacer instantanément suivant l'axe \vec{y} sans avoir besoin de changer d'orientation (*i.e.* l'angle θ , voir Figure I.3) [11]. Ces robots, souvent appelés "robots omnidirectionnels", peuvent atteindre n'importe quelle configuration dans leur environnement, en partant de n'importe quelle position, sans avoir à faire de manœuvres. Cette propriété les rend plus efficaces et plus opérationnels dans des endroits confinés, où faire des manœuvres afin de changer de direction n'est pas possible par le manque d'espace par exemple.

Cette propriété est assurée par deux types de roues, les roues orientables et les roues omnidirectionnelles [21, 25, 26]. Dans ce manuscrit, nous nous intéressons aux plateformes avec 4 roues Mecanum, nommées 4-Mecanum Wheeled Mobile Robots (4-MWMR). Plus précisément, nous considérons le robot KUKA KMP, voir Figure I.6.

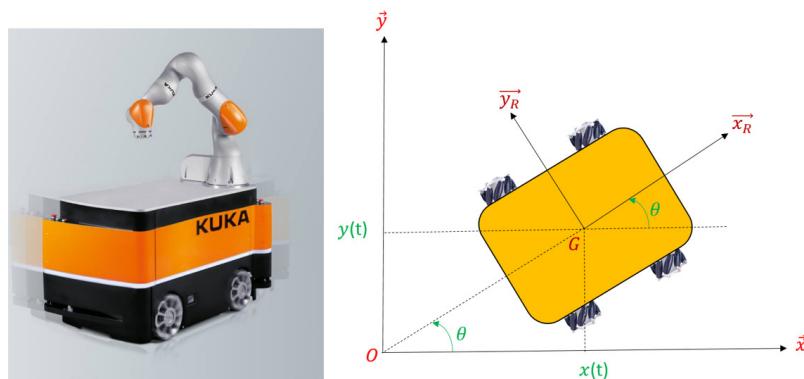


FIGURE I.6 – Robot omnidirectionnel, KUKA KMP, et sa représentation dans un plan.

La Figure I.7 résume quelques mouvements principaux que ces robots sont capables de réaliser sur une surface plane grâce aux roues Mecanum :

- Mouvement rectiligne (tout droit), toutes les roues tournent dans la même direction.
- Déplacement latéral, chaque roue se déplace dans la direction opposée de celle d'à côté.
- Déplacement en diagonal à 45°, chaque paire de roues sur une diagonale se déplace dans la même direction. La paire de roues diagonalement opposée reste immobile.
- Rotation autour de l'axe central, les roues d'un côté tournent dans le sens opposé de celles de l'autre côté.

À noter que dans tous les mouvements définis ici, toutes les roues tournent à la même vitesse. Néanmoins, la liste de ces mouvements n'est pas exhaustive. Par exemple, le mouvement en diagonal peut être effectué à 30° en actionnant une des deux roues en diagonale avec une vitesse plus élevée que la seconde.

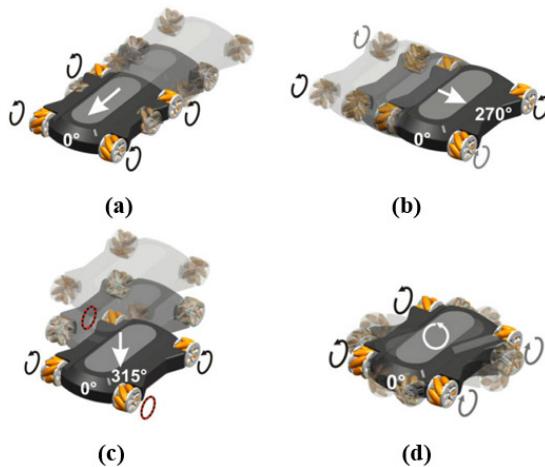


FIGURE I.7 – Quelques mouvements réalisables par le robot KUKA KMP.

I.3 Capteurs en robotique mobile

Afin d'extraire les informations utiles à la réalisation de sa tâche, il est nécessaire que le robot mobile dispose de nombreux capteurs mesurant aussi bien son état interne que l'état de l'environnement dans lequel il évolue. De ce fait, les capteurs en robotique mobile sont principalement divisés en deux catégories différentes [27] :

1. Les capteurs dits **proprioceptifs** qui permettent de mesurer l'état interne du robot (*e.g.* la vitesse de rotation des roues, la position et l'orientation du robot dans son environnement, ...).
2. Les capteurs dits **extéroceptifs** qui permettent de mesurer l'état de l'environnement d'évolution du robot (*e.g.* la température, la distance qui sépare le robot des obstacles qui l'entourent, ...).

L'instrumentation des robots mobiles n'est pas le cœur du travail de cette thèse. Nous ne citons dans cette partie que les capteurs les plus couramment utilisés dans les robots mobiles à roues d'une manière synthétique, c'est-à-dire sans détailler leurs principes de fonctionnement et sans citer leurs avantages et inconvénients. Le lecteur peut se référer à [27–33] pour plus de détails.

I.3.1 Les capteurs proprioceptifs

Mesure de la vitesse angulaire des roues : La vitesse angulaire des roues est souvent mesurée en utilisant des codeurs optiques incrémentaux, de préférence avec une résolution élevée afin de réduire le bruit de mesure.

Mesure d'orientation : L'orientation du robot qui se traduit par l'angle θ que forme le robot avec l'axe \vec{x} du repère de référence (voir Figure I.3), est généralement mesurée par un gyroscope. Plusieurs types de gyroscopes existent, les plus connus sont les gyroscopes mécaniques et optiques. Les compas et les boussoles permettent également d'avoir cette mesure.

Mesure de la vitesse de rotation du robot : La vitesse de rotation de la plateforme mobile, notée $\dot{\theta}$, est généralement mesurée à l'aide des gyromètres.

Mesure de la position du robot : Le besoin d'une localisation précise des robots mobiles dans leur environnement augmente de plus en plus, surtout dans les environnements intérieurs où l'espace de navigation des robots est restreint. En effet, la navigation autonome des robots mobiles dans leur espace de fonctionnement est basée sur le principe d'avoir une position la plus précise possible. C'est pourquoi la localisation est un sujet qui a beaucoup préoccupé, et qui préoccupe toujours les roboticiens. Plusieurs techniques ont été proposées au cours des dernières décennies afin d'améliorer de mieux en mieux la localisation des robots mobiles à roues dans leur environnement. Parmi les outils et capteurs de localisation les plus couramment utilisés, on trouve :

1. *Lodométrie* : Lodométrie consiste à intégrer dans le temps les informations provenant des capteurs de mesure de rotation des roues. C'est une technique assez simple à implémenter et à exploiter. Néanmoins, les erreurs de mesure dues au glissement par exemple sont cumulatives et rendent la mesure de plus en plus erronée. C'est pourquoi cette technique intégrée dans la plus grande majorité des robots mobiles n'est jamais utilisée seule pour mesurer la position du robot. Elle est souvent associée à d'autres techniques de localisation afin de réduire les erreurs de mesure.

2. *Le Global Positionning System (GPS)* : Le GPS est une des techniques les plus utilisées afin de déterminer la position du robot dans son environnement. En terme de précision, la mesure obtenue par GPS est entachée d'une erreur qui peut aller jusqu'à l'ordre d'une dizaine de mètres, cela rend son utilisation loin d'être une solution pour une navigation autonome, surtout dans les environnements intérieurs comme l'industrie. Par ailleurs, il existe des techniques qui peuvent améliorer cette imprécision, la plus courante étant d'utiliser deux récepteurs, dont l'un est statique et positionné avec précision dans l'environnement. Cela permet d'avoir une précision de l'ordre du centimètre [33].

D'autres outils et capteurs tels que les balises, les capteurs inertIELS, et les systèmes radar Doppler sont également beaucoup utilisés dans la localisation. Le lecteur est invité à se référer à [28–32] pour plus d'informations sur les capteurs utilisés dans la localisation.

Chaque technique de localisation a des avantages et des inconvénients. C'est pourquoi la localisation est généralement basée sur la combinaison d'approches et de plusieurs capteurs et outils redondants [34–42]. Par exemple, le filtre de Kalman est utilisé afin de combiner les données provenant de différents capteurs tel que les accéléromètres et les odomètres afin d'améliorer la localisation [43, 44].

I.3.2 Les capteurs extéroceptifs

Les capteurs servant à mesurer la distance du robot par rapport au reste des objets de son environnement (obstacles, murs, ...) sont souvent appelés **télémètres**. Il existe différentes techniques de mesure de distance, par exemple la mesure du temps de vol d'une onde ou la triangulation. De même, il existe différentes technologies pour la réalisation des télémètres. Les plus courants de ces capteurs sont [27, 33] :

1. *Capteurs à ultrason* : L'utilisation des télémètres à ultrason est très répandue dans la robotique mobile grâce à leur simplicité et leur faible coût. Ces capteurs détectent les obstacles qui se situent

dans un cône relativement large. L'avantage est la possibilité de détecter les objets fins tel que les pieds de table ou de chaise par exemple, chose qui ne peut pas être possible avec des télémètres ayant un angle d'ouverture très restreint.

2. *Capteurs à infrarouge* : Ces capteurs ont un cône de détection beaucoup plus restreint. Leur avantage est de pouvoir détecter la présence d'un obstacle et d'estimer la distance entre le robot et ce dernier avec une bonne précision.

3. *Capteurs laser* : La mesure fournie par les télémètres laser est relativement précise (avec un bruit de l'ordre de quelques centimètres) à une distance relativement grande (plusieurs dizaines de mètres). Ces derniers sont très utilisés dans des environnements intérieurs car ils fournissent des données abondantes et précises sur la position des objets caractéristiques de l'environnement tels que les murs. Néanmoins, ils possèdent une zone de perception limitée à un plan. Ils ne permettent donc pas de détecter les obstacles situés hors de ce plan. Ils ne peuvent pas non plus détecter les objets ne réfléchissant pas correctement la lumière du laser. Pour limiter ces inconvénients, ils sont généralement utilisés en conjonction avec des capteurs à ultrason qui ont un cône de détection beaucoup plus large.

I.3.3 Les caméras

L'utilisation des caméras pour la navigation des robots mobiles est une approche très intéressante car elle permet à la fois de localiser le robot (avoir sa position) et de détecter les objets qu'il y a autour, ainsi que d'estimer leur distance par rapport au robot. Le seul bémol des approches à base des caméras est le traitement des données volumineuses et complexes fournies par ces capteurs, qui est une tâche difficile [27].

Il est à souligner que la liste des capteurs donnée dans cette section est loin d'être exhaustive. Nous pouvons aussi citer les capteurs de température, de la charge de batterie, ...

I.4 Terminologie et généralités

I.4.1 Terminologie sur le diagnostic

La terminologie du diagnostic et de la détection des *défauts* utilisée dans les travaux de cette thèse a été initialement proposée par les Prof. *Isermann & Ballé* [45] et adoptée par le comité technique de IFAC ‘SAFEPROCESS’, afin de standardiser la terminologie du domaine du diagnostic. Cette même terminologie a été reproduite par la suite dans l’Annexe B de [46].

Pour commencer, qu'est-ce que c'est qu'un *défaut*? Selon [45], un *défaut* peut être défini comme étant une déviation d'un ou de plusieurs paramètres du système de sa (leur) valeur nominale, et il peut être annonciateur de défaillance.

La défaillance se traduit par une interruption permanente de la capacité d'un système pour exécuter une fonction requise dans des conditions de fonctionnement spécifiées.

La détection d'un *défaut* (Fault Detection) consiste à signaler et à mettre en évidence la présence du *défaut* ainsi que son instant d'apparition.

L'isolation d'un *défaut* (Fault Isolation) consiste à déterminer et à localiser l'élément défaillant dans le système.

L'identification d'un *défaut* (Fault Identification) consiste à estimer son amplitude et à prédire son évolution temporelle.

Le processus de détection et d'isolation des *défauts* est souvent mis sous l'acronyme 'FDI' pour 'Fault Detection and Isolation', tandis que l'acronyme 'FDD' pour 'Fault Detection and Diagnosis' est utilisé pour parler du diagnostic, soit la détection, l'isolation, et l'identification des *défauts*.

La surveillance (Monitoring) est définie comme étant une tâche continue en temps réel, visant à déterminer l'état de fonctionnement d'un système physique, en enregistrant les informations, et en reconnaissant et signalant les anomalies de son comportement.

Enfin, la supervision (Supervision) d'un système consiste à le surveiller et à prendre des actions appropriées afin de maintenir son fonctionnement en cas de *défaut*.

I.4.2 Généralités sur les *défauts* et le diagnostic

I.4.2.1 Classification des *défauts*

Pour classifier un *défaut*, plusieurs critères sont pris en compte [47–50], notamment : 1) En fonction de l'élément du système qu'il affecte, et 2) Selon son évolution temporelle.

Ainsi, selon le premier critère, un *défaut* apparaît généralement sur trois niveaux du système (voir Figure I.8 suivante) ;

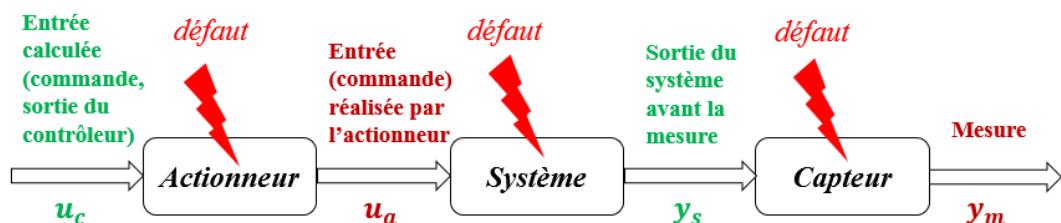


FIGURE I.8 – Différents niveaux d'apparition de *défauts*.

- Le *défaut* actionneur représente un dysfonctionnement d'un dispositif agissant sur la dynamique du système tel qu'un moteur. Parmi les *défauts* actionneurs les plus connus, on peut citer ; le blocage, la saturation, et la perte d'efficacité (voir Figure I.9).

Ces *défauts* peuvent être présentés par les équations suivantes [47], où $t_{\text{défaut}}$ désigne l'instant d'apparition du *défaut*, $u_{c,i}$ et $u_{a,i}$ représentent respectivement l'entrée calculée et exécutée par le $i^{\text{ème}}$ actionneur, et enfin $k_i(t)$ représente l'efficacité de l'actionneur i et ϵ_i l'efficacité minimale

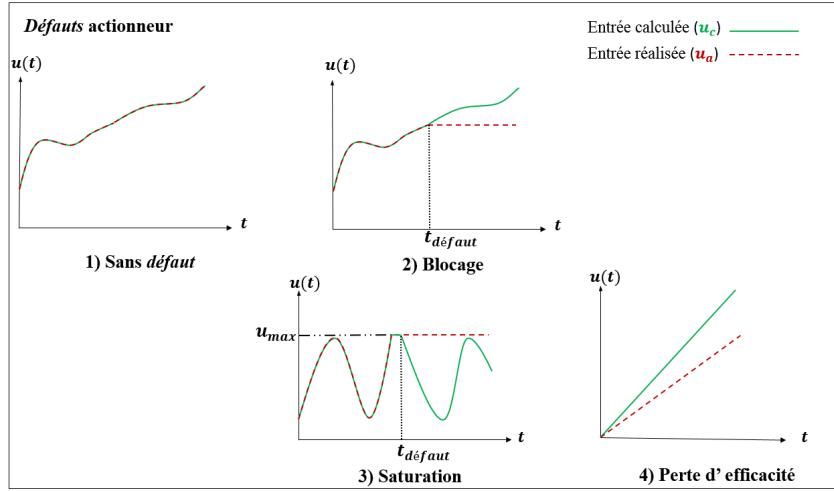


FIGURE I.9 – Quelques défauts actionneur.

de l'actionneur i .

1. Absence de *défaut* : $u_{a,i}(t) = u_{c,i}(t); \forall t > t_0$
 2. Blocage : $u_{a,i}(t) = u_{c,i}(t_{\text{défaut}}); \forall t > t_{\text{défaut}}$
 3. Saturation : $u_{a,i}(t) = \mu_{c,i}; \forall t > t_{\text{défaut}}$, avec $\mu_{c,i}$ est la valeur minimale ou maximale de la saturation
 4. Perte d'efficacité : $u_{a,i}(t) = k_i(t)u_{c,i}(t); 0 < \epsilon_i < k_i(t) < 1; \forall t > t_{\text{défaut}}$
- Le *défaut* système se traduit par le changement des paramètres internes du système, entraînant une modification de sa dynamique, comme une redéfinition inattendue des coefficients aérodynamiques [50].
 - Le *défaut* capteur se traduit par une variation anormale d'une mesure. Parmi les *défauts* capteurs les plus répandus, on peut citer le biais, la dérive, le blocage, et le *défaut* de calibrage (voir Figure I.10).

Ces *défauts* peuvent être présentés par les équations suivantes [47], où $t_{\text{défaut}}$ est l'instant d'apparition du *défaut* du $i^{\text{ème}}$ capteur, $b_i(t)$ la précision du $i^{\text{ème}}$ capteur avec $b_i \in [-\bar{b}_i, \bar{b}_i]$ tel que $\bar{b}_i > 0$ est la précision minimale du capteur i , et enfin $k_i(t)$ représente l'efficacité du $i^{\text{ème}}$ capteur avec $k_i(t) \in [\bar{k}_i, 1]$ et $\bar{k}_i > 0$ sachant que \bar{k}_i est l'efficacité minimale du capteur i .

1. Absence de *défaut* : $y_{m,i}(t) = y_{s,i}(t); \forall t > t_0$
2. Biais : $y_{m,i}(t) = y_{s,i}(t) + b_i; \dot{b}_i = 0; b_i(t) \neq 0; \forall t \geq t_{\text{défaut}}$
3. Dérive : $y_{m,i}(t) = y_{s,i}(t) + b_i(t); |b_i(t)| = c_i t; 0 < c_i \ll 1; \forall t \geq t_{\text{défaut}}$
4. Blocage : $y_{m,i}(t) = y_{m,i}(t_{\text{défaut}}); \forall t \geq t_{\text{défaut}}$
5. *Défaut* de calibrage : $y_{m,i}(t) = k_i(t)y_{s,i}(t); 0 < \bar{k}_i \leq k_i(t) \leq 1; \forall t \geq t_{\text{défaut}}$

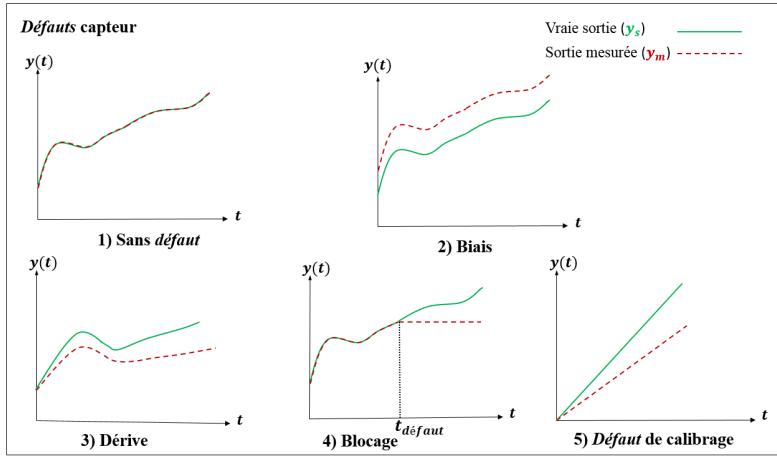


FIGURE I.10 – Quelques défauts capteur.

Pour le second critère (*i.e.* selon l'évolution temporelle du *défaut*), et selon [48], un *défaut* peut être abrupt ou graduel (*i.e.* progressif, dérive) (Abrupt or Drift-like Faults), voir Figure I.11.

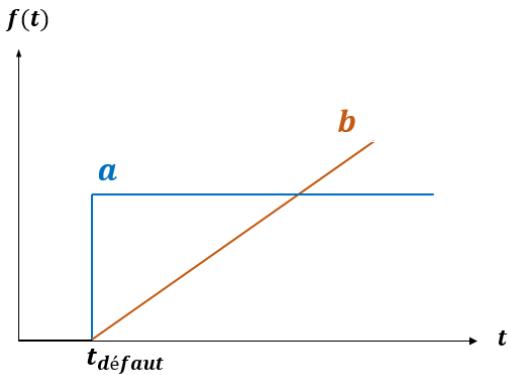


FIGURE I.11 – a) Défaut abrupt, b) Défaut graduel.

De plus, le *défaut* peut être permanent (*c'est-à dire* que le *défaut* subsiste tant qu'une opération de maintenance corrective n'a pas été effectuée), ou intermittent (*c'est-à dire* que le *défaut* persiste sur une durée déterminée et limitée. Après cette durée, le dispositif est apte à assurer la fonction ou la mission pour laquelle il a été conçu sans avoir fait l'objet d'une action corrective) [47] (voir Figure I.12).

D'autres critères peuvent également être pris en compte afin de classifier les *défauts* [48]. Par exemple, un *défaut* peut être additif ou multiplicatif (Additive or Multiplicative Faults), voir Figure I.13). Un *défaut* additif est un signal parasite qui vient se superposer en un point sur le schéma fonctionnel, tandis que généralement les *défauts* multiplicatifs induisent des changements sur la corrélation du signal de sortie du système ainsi que des changements dans sa dynamique [50].

Quelque soit son type, un *défaut* s'il n'est pas détecté, isolé, et accommodé à temps, il peut conduire à la défaillance du composant et à la perte de la controllabilité du système entier.

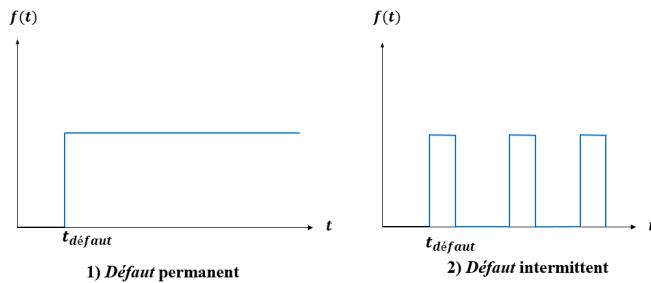


FIGURE I.12 – Défaut permanent et intermittent.

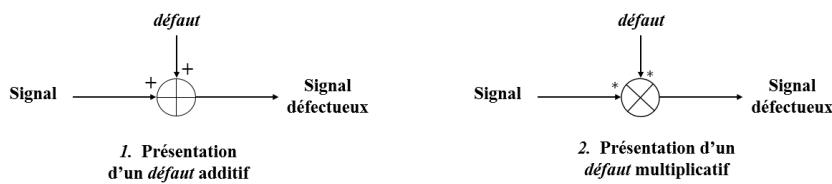


FIGURE I.13 – Défaut additif et multiplicatif.

C'est pour cette raison que les systèmes autonomes sont de plus en plus dotés d'un système intelligent leur permettant de détecter et de signaler leurs *défauts* internes. La détection précoce de tout dysfonctionnement (*i.e.*, *défaut*) permet d'éviter tout dégât pouvant être engendré suite à ce dernier, en planifiant une maintenance corrective afin de réparer ou de remplacer le composant défaillant par exemple. C'est le but du diagnostic.

I.4.2.2 Classification des méthodes de diagnostic

Pour classifier une méthode de diagnostic (*i.e.* détection, isolation, et identification des *défauts*), différents critères sont pris en considération : la dynamique du système (discrète, continue, ou hybride), sa complexité, l'implémentation de diagnostic en ligne et/ou hors ligne, la nature d'information (qualitative et/ou quantitative) et sa profondeur (structurelle, fonctionnelle et/ou temporelle), ... Dans ce contexte, plusieurs classifications sont proposées dans la littérature [51–57]. Ces classifications sont influencées par les terminologies et les contextes particuliers de chaque communauté et ne sont donc pas toujours homogènes [50].

L'objectif dans cette thèse est de choisir la méthode la plus appropriée pour la résolution de notre problème du diagnostic. Il est important de positionner la classe des méthodes de diagnostic qui nous intéresse parmi les différentes méthodes de la littérature. Ici, l'élément discriminant quant au choix de l'approche est la connaissance approfondie ou non de la physique du système. Ainsi, deux classes principales des méthodes de diagnostic sont distinguées (voir Figure I.14) :

- 1 Les approches dites "à base de données" (Data-based methods) ; ou sans modèle mathématique. Elles ne nécessitent pas une connaissance détaillée du système, mais elles utilisent des connaissances non liées à la physique du système.

2 Les approches dites "à base de modèle" (Model-based approaches). Ces approches nécessitent une connaissance approfondie de la physique du système.

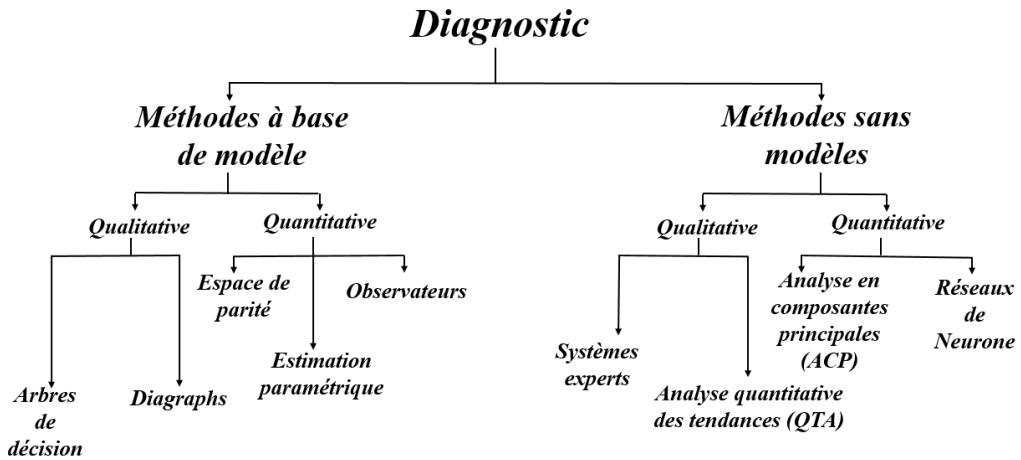


FIGURE I.14 – Classification de quelques approches de diagnostic.

Comme illustré dans la Figure I.14, les approches de ces deux classes peuvent être quantitatives ou qualitatives. Un overview des méthodes qualitatives et quantitatives à base de données peut être trouvé dans [58], tandis qu'un review des approches quantitatives et qualitatives à base de modèles peut être trouvé dans [55, 59] respectivement. Une analyse comparative de quelques unes de ces méthodes peut être trouvée dans [50].

Les méthodes basées sur les données (Data-based methods) nécessitent la collecte et la manipulation d'une grande quantité de données sous le mode de fonctionnement nominal et dégradé du système, tandis que celles basées sur un modèle (Model-based methods) nécessitent généralement la modélisation du comportement du système sous forme d'équations mathématiques. En raison du manque de données, les approches utilisant le modèle mathématique du système sont un bon compromis pour le cas des travaux de cette thèse. De plus, selon [51], les méthodes à base de modèles sont plus performantes que celles basées sur le traitement de données ou sur les approches relationnelles. Les avantages des approches à base de modèle sont :

- Le faible coût de développement et de maintenance.
- Les défauts n'ont pas besoin d'être anticipés.
- Il s'agit des connaissances de conception plutôt que d'exploitation.
- La connaissance du système est découpée de la connaissance de diagnostic.
- Seule l'information du comportement normal (*i.e.* sans défauts) est prise en compte.

Généralement, les approches "qualitatives" à base de modèle utilisent la connaissance et l'expérience (expert-based knowledge methods). Elles sont basées sur les relations de causes à effets [59], tandis que les approches "quantitatives" sont basées sur la génération d'équations de redondance analytique donnant naissance à ce qu'on appelle "les résidus" [55].

Dans le cadre de cette thèse, nous disposons de peu de connaissance sur le comportement des robots étudiés sous les différents modes de dégradation, notre attention va alors se porter sur les approches quantitatives. Parmi les approches quantitatives à base de modèle, nous pouvons citer les méthodes fondamentales et très utilisées dans la littérature [50] :

- Espace de parité : le principe de cette approche est de vérifier la cohérence entre le modèle du système et les mesures issues des capteurs et des entrées connues (sorties du contrôleur, commandes). Les premières utilisations de cette approche ont été réalisées à partir des relations analytiques statiques [60, 61]. Ces notions ont été généralisées ensuite dans [62], puis dans [63]. D'une façon générale, la méthode consiste à réaliser une redondance analytique entre les entrées et les sorties du système, indépendamment de ses états internes. La matrice de parité est définie comme étant une matrice de projection de la matrice d'observabilité, ce qui revient à éliminer l'influence des états sur les résidus.
- Estimation paramétrique : Le principe de cette approche est basé sur l'estimation des paramètres du système en continu et en temps réel, en utilisant les mesures d'entrées et de sorties, et de calculer les résidus qui correspondent à la différence entre ces estimations et les valeurs de référence de l'état normal du système. Les premières applications de cette approche ont été faites dans [64]. De nombreuses applications de cette technique ont été réalisées dans le domaine de l'automobile [65, 66], ou pour un modèle non-linéaire d'un satellite [67].
- Observateurs : Le principe de base du diagnostic par observateur consiste à estimer la sortie du système à partir de la connaissance de la commande (ou partie de la commande), et des mesures (ou partie des mesures). Le vecteur résidu est ainsi construit comme l'écart entre l'entrée ou la sortie estimée et l'entrée ou la sortie mesurée. Ces résidus doivent servir d'indicateurs fiables du comportement du processus. Il existe dans la littérature plusieurs approches permettant de concevoir un observateur pour les systèmes linéaires et non-linéaires. Les premières techniques ont été d'abord proposées pour les systèmes linéaires invariants dans le temps (LTI) avec le filtre de Kalman [68] et l'observateur de Luenberger [69]. Le filtre de Kalman jouit d'une très grande popularité notamment dans l'industrie grâce à sa simplicité d'implémentation [70], mais malheureusement son application aux systèmes non-linéaires ne garantit pas toujours une convergence asymptotique de l'erreur d'observation. Les chercheurs se sont ensuite intéressés aux systèmes non-linéaires et plusieurs approches de conception des observateurs ont été proposées. L'un des points de départ de l'étude de la synthèse d'observateurs pour le cas non-linéaire est la linéarisation de l'erreur d'observation [71–76]. Cette approche est basée sur la description du système non linéaire sous une forme où le système est affine selon l'état et où les non-linéarités ne dépendent que de l'entrée et de la sortie par un changement de coordonnées approprié, puis la synthèse d'un observateur de Luenberger. Cependant, la recherche d'une telle transformation permettant de linéariser l'erreur d'observation n'est pas forcément facile. D'autres approches ont été également proposées, comme par exemple les méthodes qui permettent de transformer un problème d'observabilité en un problème basé sur la résolution des LMI [77],

et l'approche basée sur le calcul d'intervalles [78, 79]. D'un autre côté, les observateurs de Luenberger et de Kalman ont été également étendus et adaptés aux cas non-linéaires [80–82]. A côté des deux observateurs évoqués dans ce paragraphe, on peut trouver plusieurs autres types tels que les observateurs à grand gain [83], les observateurs à entrées inconnues [84], ...

I.4.2.3 Les résidus

Comme mentionné précédemment, les résidus peuvent être définis comme la différence entre les entrées calculées ou les sorties mesurées d'un système, et leurs estimations obtenues en utilisant le modèle mathématique du système. Cette définition est valable pour le cas de la redondance analytique (*i.e.* en utilisant le modèle du système). Néanmoins, la redondance analytique n'est pas le seul moyen de génération de résidus. En effet, dans certains systèmes critiques (*e.g.* avions, satellites, centrales nucléaires, ...), afin de détecter spécifiquement les *défauts* capteurs, on utilise plus d'un seul capteur (un nombre impair de capteurs, au minimum trois) pour la même mesure [85]. Ainsi, la différence entre la mesure de chaque deux capteurs génère un résidu. Cette approche est appelée redondance matérielle.

Le principe d'isolation des *défauts* avec la redondance matérielle est illustré dans le schéma de la Figure I.15 et la Table I.1, où $C_i, i = 1, \dots, 3$ dénote le $i^{\text{ème}}$ capteur.

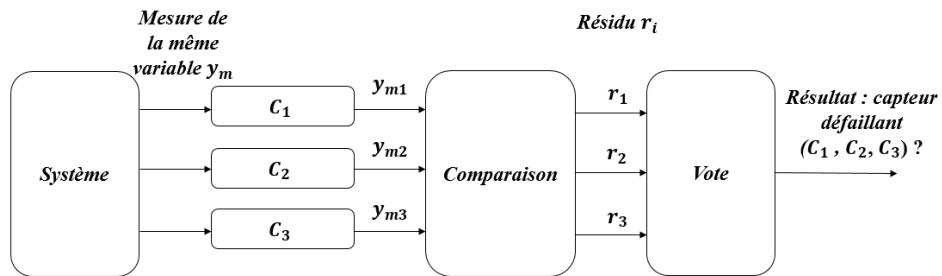


FIGURE I.15 – Principe d'utilisation de la redondance matérielle.

TABLE I.1 – Principe d'isolation des *défauts* capteur en utilisant la redondance matérielle.

Capteur défaillant	Signature de $r_1 = y_{m1} - y_{m2}$	Signature de $r_2 = y_{m1} - y_{m3}$	Signature de $r_3 = y_{m2} - y_{m3}$
C_1	1	1	0
C_2	1	0	1
C_3	0	1	1

Généralement et dans le cas idéal, les résidus (qu'ils soient générés par la redondance analytique ou matérielle) sont nuls en absence de *défauts*. Mais quand un *défaut* apparaît, les résidus lui étant sensibles s'éloignent de zéro, signalant ainsi la présence de ce *défaut*. Dans les applications réelles, les *défauts* ne sont pas les seuls qui peuvent impacter les résidus. En effet, les systèmes peuvent être sujets au bruit de mesure et à des perturbations, ce qui peut avoir un impact sur les résidus et conduire

à détecter des fausses alarmes (*i.e.* signaler la présence d'un *défaut* alors qu'en vérité il n'y en a pas). Par conséquent, les algorithmes de détection et d'isolation des défauts (FDI Algorithms) doivent être conçus pour être sensibles le moins possible aux bruits et aux perturbations. Généralement, deux étapes principales sont utilisées pour traiter les fausses alarmes [86] :

- 1) Le filtrage des résidus (Residual Filtering).
- 2) Le test statistique des résidus (Residual Statistical Testing) en imposant des seuils. En effet, lorsqu'un résidu dépasse son seuil prédéfini, il signale la présence d'un *défaut*.

La première étape est généralement utilisée lorsque le bruit de mesure est très important. Il vise à réduire l'effet du bruit sur les résidus. Concernant la deuxième étape, les seuils peuvent généralement être fixes ou adaptatifs [87, 88]. Les seuils adaptatifs sont principalement dédiés aux systèmes soumis à des incertitudes et à des perturbations [88]. Dans les travaux de cette thèse, en prenant en compte les conditions de fonctionnement dans la fab (*i.e.* surface plane sans pente, pas de glissement, des frottements négligeables, ...), les obstacles sont les seuls éléments qui peuvent perturber les robots. Sachant que les obstacles sont pris en compte dans la navigation des robots, nous supposons qu'il n'y a pas de perturbations et que le bruit de mesure est le seul problème auquel sont confrontés les robots.

Concernant les bruits de mesures, nous supposons connaître la précision des capteurs (*i.e.* la dispersion des mesures autour de la mesure réelle). Cette précision est généralement fournie par le constructeur. Par conséquent, le bruit de mesure de chaque capteur est modélisé en prenant en considération sa précision. En tenant compte de toutes ces remarques, les seuils fixes sont les mieux adaptés à nos approches de diagnostic. Pour fixer ces seuils, la règle des 3σ est utilisée [89].

Il est important de noter qu'en utilisant la règle de 3σ , 99.73% des mesures sont sauvegardées. Le reste (0.27%) est rejeté et considéré comme de fausses informations conduisant à générer de fausses alarmes.

Afin d'être robuste à ces fausses alarmes, pour chaque résidu généré r_i , les résidus de décision, notés R_i , qui reflètent la signature des résidus r_i (0 en absence de *défaut* et 1 en sa présence), sont assignés à 1 pour détecter la présence d'un *défaut* si et seulement si le résidu r_i correspondant dépasse son seuil prédéfini (*i.e.* predefined thresholds, noté $r_{th,i}$) pendant un certain laps du temps, noté NT_e , T_e étant la période d'échantillonnage et N est à déterminer expérimentalement, de telle sorte qu'il y ait un compromis entre la réduction de fausses alarmes et la rapidité de détection des vrais *défauts*. Par conséquent, l'algorithme de détection suivant est utilisé :

$$R_i(k) = \begin{cases} 1 & \text{si } |r_i(j)| > r_{th,i}, \forall j \in [k-N+1, k] \\ 0 & \text{sinon} \end{cases} \quad (\text{I.1})$$

I.4.2.4 Banc d'observateurs

La méthode la plus simple pour l'isolation des *défauts* est généralement basée sur la génération d'un nombre de résidus ayant une table de signatures caractérisant chaque *défaut*. Afin d'avoir suffisamment de résidus et faciliter l'étape d'isolation, on utilise généralement ce qu'on appelle les "bancs d'observateurs" (Observer Bank), dont on distingue deux structures différentes : 1) la structure généralisée (Generalized Observer Structure, ou *GOS*) introduite par *Frank* [90], et 2) la structure dédiée (Dedicated Observer Structure, ou *DOS*), introduite par *Clark* [91].

Généralement, on distingue deux cas de figure selon si l'on étudie les *défauts* capteurs ou actionneurs. Dans le cas des *défauts* actionneurs, et comme illustré sur la Figure I.16, la structure *GOS* consiste à mettre à l'entrée de chaque observateur i tout le vecteur des sorties mesurées Y_m , plus tout le vecteur d'entrée u_c à l'exception d'une seule entrée $u_{c,i}$. En effet, la sortie de chaque observateur i sera sensible aux *défauts* de tous les actionneurs à l'exception de l'actionneur dont l'entrée est $u_{c,i}$. Inversement, dans le cas de la structure *DOS*, chaque observateur i a en entrée tout le vecteur des sorties Y_m plus une seule entrée $u_{c,i}$ à chaque fois. Dans ce cas, chaque observateur i ne sera sensible qu'aux *défauts* de l'actionneur dont l'entrée est $u_{c,i}$.

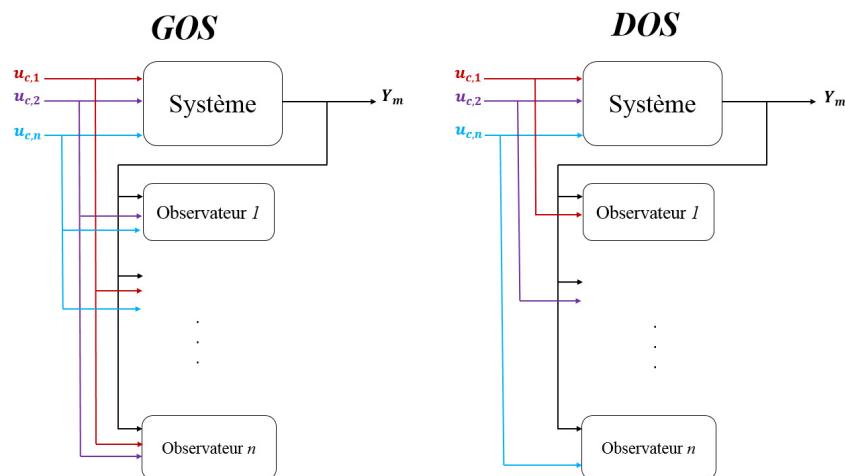


FIGURE I.16 – Structures *GOS* et *DOS* dans le cas des *défauts* actionneurs.

Également, dans le cas des *défauts* capteurs, et selon la Figure I.17, la structure *GOS* consiste à mettre à l'entrée de chaque observateur tout le vecteur d'entrée u_c , et tout le vecteur des sorties (*i.e.* toutes les mesures) à l'exception de la mesure $y_{m,i}$. Ainsi, la sortie de chaque observateur va être sensible à tous les *défauts* capteurs à l'exception du capteur dont la sortie est $y_{m,i}$. A l'inverse, dans le cas de la structure *DOS*, chaque observateur i a en entrée tout le vecteur d'entrée u_c plus une seule sortie $y_{m,i}$. Dans ce cas, la sortie de chaque observateur va être sensible uniquement aux *défauts* du capteur dont la mesure est $y_{m,i}$. Il est à noter que la notion d'observabilité doit être vérifiée afin de pouvoir créer les bancs d'observateurs.

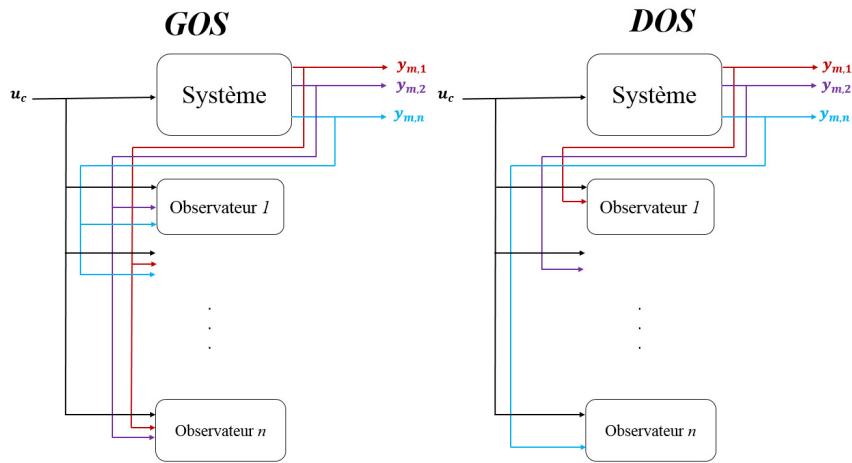


FIGURE I.17 – Structures *GOS* et *DOS* dans le cas des *défauts* capteurs.

I.5 Bibliographie sur le diagnostic des robots mobiles à roues

Le diagnostic des *défauts* des robots mobiles à roues une thématique qui a été largement abordée dans la littérature, et qui est toujours d'actualité. Dans cette partie, nous proposons une vue d'ensemble des travaux proposés par les chercheurs à travers les années afin de répondre à cette problématique.

Différentes approches à base de data ou du modèle ont été proposées afin de diagnostiquer les *défauts* que ce soit au niveau des actionneurs, du système, ou des capteurs des robots mobiles à roues. Dans le paragraphe qui suit, nous explorons les principaux travaux de recherches effectués dans ce domaine, en nous intéressant plus particulièrement aux robots unicycles et omnidirectionnels. L'abréviation WMRs est utilisée afin de parler des robots mobiles à roues (Wheeled Mobile Robots).

Nous pouvons commencer par citer un review [92] englobant quelques travaux proposés dans la littérature sur le diagnostic et la commande tolérante aux fautes des WMR. Les auteurs dans [93] divisent les WMRs en sous-systèmes (sous-système d'alimentation, sous-système de conduite, sous-système de direction, sous-système de suspension qui comprend les roues et les suspensions, sous-système de communication qui permet l'échange des signaux de commande et des données avec le centre de contrôle ou avec d'autres robots, et les capteurs), puis ils proposent une synthèse de quelques travaux réalisés sur le diagnostic de chaque sous-système, particulièrement le diagnostic du sous-système de conduite, des capteurs, du sous-système de communication, et du sous-système de suspension. De plus, ils s'intéressent au pronostic et ils proposent une synthèse de deux travaux publiés dans la littérature afin de prédire la durée de vie utile restante (Remaining Useful Life ou RUL) de la batterie et du sous-système de suspension.

Dans [94], les auteurs s'intéressent à quatre méthodes à base de modèle utilisées pour le diagnostic des *défauts* des WMRs, notamment l'espace de parité (parity space), le modèle de Markov caché (Hidden Markov Model ou HMM), le filtre particulier (particle filter), et le modèle d'opérateur observable (Observable Operator Model ou OOM), et proposent une synthèse et une analyse comparative de ces méthodes.

Dans [95], une approche basée sur l'intelligence artificielle a été développée afin de construire un système de supervision pour le diagnostic et l'accommodation des *défauts* des WMRs. Les auteurs dans [96] ont testé et discuté l'efficacité de la technique de "la longueur minimale du message" (minimum message length ou MML) qui est une technique d'apprentissage probabiliste, pour le diagnostic des *défauts* d'un ou de plusieurs composants des WMRs simultanément (*e.g.* caméra, unité d'inclinaison de la caméra, le panoramique de la caméra, ...). Les résultats montrent que cette technique ne peut pas être adaptée aux WMRs et la probabilité que l'état du système soit correctement diagnostiqué est moins de 1%.

Les auteurs dans [97] considèrent les *défauts* au niveau des roues (*i.e.* roue bloquée, défaillance du moteur ou du servo-moteur) des robots unicycles. Ils proposent d'utiliser une approche de modélisation à base des réseaux de neurones (*i.e.* Local Model Neural Network ou LMNN) afin de modéliser le comportement de chaque roue et de générer des résidus qui sont dans ce cas la différence entre les sorties du robot et les estimations du modèle de chaque roue. Ensuite, le bruit de mesure de ces résidus est filtré en utilisant des algorithmes visant à lisser la courbe du comportement des résidus (*i.e.*, change detection algorithms [98]) afin de diagnostiquer les *défauts* considérés.

Les *défauts* au niveau de l'entraînement d'un robot de football omnidirectionnel sont considérés dans [99], et une approche hybride prenant en compte les états discrets de fonctionnement du robot et de son déplacement en continu est proposée afin de les détecter et de les compenser.

Dans [100], les *défauts* du logiciel de contrôle (Control software) des WMRs sont étudiés et une méthode à base d'observateurs est proposée afin de les détecter et de les isoler, tandis que pour les mêmes *défauts*, les auteurs dans [101] proposent une approche à base des réseaux de neurones. Les auteurs dans [102] considèrent trois types de *défauts* des robots unicycles (*i.e.* *défauts* des roues gauche/droite et le changement des paramètres dynamiques du robot) et proposent une approche à base d'un observateur non-linéaire pour diagnostiquer ces *défauts*. Dans [87], les auteurs utilisent une linéarisation locale des modèles des WMRs avec une technique de modélisation de l'erreur afin de générer des seuils adaptatifs, donnant naissance à une nouvelle approche du diagnostic des WMRs.

Afin d'étudier les *défauts* capteurs, les auteurs dans [103] proposent d'utiliser une estimation adaptative à plusieurs modèles basée sur la technique du filtre de Kalman (KF), tandis que la même technique est combinée avec les réseaux de neurones dans [104]. D'un autre côté, un banc d'observateurs de Kalman est combiné avec les systèmes experts dans [105], tandis que dans [106], le filtre de Kalman basé sur la technique d'identification est utilisé pour traiter le problème d'isolation des *défauts*. Afin de diagnostiquer les *défauts* du système de navigation à l'estime (Dead-reckoning system), notamment des codeurs optiques et du gyroscope, les auteurs dans [107] proposent de combiner les règles d'inférence (rule-based inference framework) avec un banc de filtres particulaires (particle filters), tandis que dans [108], les mêmes *défauts* sont considérés et une approche multi-modèles est utilisée pour les diagnostiquer. Les auteurs dans [109] considèrent différents types de *défauts* capteur (*i.e.* défaillance, bruit, *défaut* de calibrage) et utilisent la théorie du filtre de Kalman pour les détecter et les isoler.

Dans [110], une approche bayésienne est adaptée afin de détecter les *défauts* capteurs d'un robot à 4 roues, et dans [111], le filtre de Kalman est combiné avec une méthode de classification floue (fuzzy cluster) afin de détecter les *défauts* simultanés des capteurs d'un robot à 3 roues.

Concernant les *défauts* actionneurs, les robots unicycles sont considérés dans [112], et une méthodologie basée sur l'analyse structurale (structural analysis-based technique) est utilisée afin de les détecter. Dans [113], un banc d'observateurs de Luenberger est utilisé. Les auteurs s'appuient ensuite sur un arbre de décision pour diagnostiquer trois types de *défauts* actionneurs, à savoir la dégradation des performances des actionneurs, la non-réponse des actionneurs (unresponsive actuators), et le blocage des actionneurs (Stuck-at-Zero) des robots unicycles, tandis que les auteurs dans [114] proposent une approche à base d'un banc d'observateurs de Kalman étendus (Extended Kalman Observers) pour résoudre le même problème.

Pour le diagnostic des *défauts* capteurs et actionneurs à la fois, une nouvelle approche utilisant la corrélation entre les informations disponibles sur le robot et leur relation hiérarchique est proposée dans [115], tandis que les auteurs dans [116] proposent une méthodologie basée sur un banc d'observateurs.

I.6 Bibliographie sur la navigation des robots mobiles à roues

Dans cette partie, nous allons donner une vue générale sur la navigation des robots mobiles à roues afin d'introduire la seconde problématique traitée dans ce manuscrit : gestion du retard généré par les obstacles mobiles. La gestion du retard fait partie de la navigation.

Généralement, les robots évoluent dans deux types d'environnements différents : 1) Les environnements statiques, et 2) Les environnements dynamiques. Un environnement est dit "statique" quand il n'évolue pas dans le temps, et lorsqu'il ne contient que des obstacles fixes et immobiles (e.g. les murs, des équipements fixes, ...). Inversement, il est dit "dynamique" lorsqu'il évolue constamment, et lorsqu'il contient des obstacles mobiles à côté des obstacles fixes (e.g. tables, chaises, humains, ...). Comparés aux environnements statiques, la navigation dans les environnements dynamiques est plus compliquée à réaliser car les robots doivent s'adapter à l'évitement d'obstacles imprévus et aux changements imprédictibles de cet environnement.

Afin de pouvoir réaliser une tâche d'une façon autonome dans leur environnement d'évolution (i.e. sans aucune intervention ou assistance humaine), les robots mobiles à roues doivent être capables de se localiser, de suivre l'évolution de leur environnement (perception), de générer un chemin de leur position initiale à leur destination (planification de chemin ou path planning), de suivre ce chemin d'une manière efficace (génération et suivi de trajectoire), et de respecter les conditions de travail [117]. C'est l'objectif principal de la planification du mouvement (motion planning) et de la commande.

La planification du mouvement est un problème bien connu dans la littérature. En effet, un nombre important de travaux de recherche a été consacré à cette problématique au fil des années, et

a donné naissance à différentes méthodes visant à améliorer de plus en plus la navigation autonome des robots mobiles. En règle générale, la résolution d'un problème de planification du mouvement repose sur une approche de décomposition [118] : a) Résolution du problème de planification de chemin (path planning), et b) Génération d'une trajectoire conforme au chemin planifié (trajectory generation) et l'utilisation d'une loi de contrôle pour suivre cette dernière (trajectory tracking). De nombreux ouvrages importants englobant les approches de planification du mouvement proposées au fil des années peuvent être trouvés dans la littérature sous forme de livres pour la plupart. Parmi ces références, on peut citer [119–123].

a) Planification du chemin (Path planning)

La planification du chemin prend en compte le problème d'évitement d'obstacles et vise à trouver un chemin optimal ou sous-optimal sans collision (optimal or sub-optimal free-collision path), permettant au robot d'atteindre une position désirée à partir d'une configuration initiale dans son environnement. D'une manière générale, le problème de la planification de chemin a été abordé de deux manières différentes, en utilisant deux types de méthodes, appelées méthodes "globales" et "locales" [122, 124].

Les approches globales, dites également "approches délibératives", supposent la connaissance préalable et complète de l'environnement du robot. Elles sont basées sur la carte (map) de l'espace de travail du robot (obtenue à partir d'une étape de cartographie où la géométrie des zones de travail et les obstacles statiques sont modélisés). Une liste non-exhaustive des approches délibératives de planification de chemin peut être donnée comme suit :

1. Les Roadmaps (tels que les graphes de visibilité (Visibility Graph), le diagramme de Voronoï (Voronoi Graph) [125] et les graphes de connectivité (Connectivity Graph) [126]).
2. Les méthodes de décomposition en cellules (Cell-Decomposition methods) [122] (telles que critical-Curve Based Decomposition et Rectanguloid Cell Decomposition)
3. Les méthodes de champ potentiel artificiel [127] (telles que les champs de potentiel avec descente de gradient et fonctions de potentiel harmonique, Potential Field with Gradient Descent and Harmonic Potential Functions).

Ces approches délibératives ont montré leur efficacité dans différentes situations mais elles souffrent d'un inconvénient majeur résidant dans l'hypothèse de la connaissance complète de l'environnement de travail du robot. Cela limite leur utilisation aux environnements statiques [128].

Quant aux approches locales, appelées également «approches réactives», elles ne nécessitent pas la connaissance complète de l'environnement, contrairement aux méthodes délibératives. Elles sont basées sur l'utilisation des données des capteurs proprioceptifs uniquement (e.g. télémètres laser, caméras, capteurs à ultrason, odomètres, ...), et elles utilisent les connaissances locales de l'environnement pour planifier le chemin. Une liste non-exhaustive des approches locales de navigation peut être définie comme suit :

1. La logique floue [129].
2. Les réseaux de neurones [130].

3. Les algorithmes génétiques [131].
4. L'optimisation des colonies de fourmis [132].

A noter que plus de méthodes peuvent être trouvées dans [133]. Ces approches sont parfaitement adaptées aux environnements dynamiques, mais elles souffrent de certaines faiblesses majeures comme : 1) elles ne peuvent pas garantir un chemin optimal, et 2) il peut y avoir des situations où l'algorithme ne peut pas trouver la position cible (voir [134]). Une vue d'ensemble détaillant les approches locales et globales les plus utilisées dans la navigation autonome des robots mobiles peut être trouvée dans [135].

Pour remédier aux inconvénients des approches délibératives et réactives, de nouvelles approches hybrides ont vu le jour dans les années 1990 [136]. Elles visent à combiner les avantages des approches locales et globales et à éliminer certaines de leurs faiblesses. Le lecteur est invité à consulter les références citées précédemment pour plus de détails sur les approches de planification de chemins évoquées dans ce paragraphe.

b) Génération de trajectoire (Trajectory generation)

Après avoir défini un chemin sans collision (free-collision path) pour aller d'une position initiale à une position désirée dans l'environnement du robot, l'étape suivante de la planification du mouvement consiste à générer une trajectoire conforme à ce chemin. Un des objectifs fondamentaux est de générer des trajectoires en tenant compte de plusieurs critères et contraintes tels que la distance à parcourir et l'instant d'arrivée prévu [137]. Dans le paragraphe qui suit nous allons citer quelques travaux de recherche traitant la thématique de génération de trajectoire.

Dans [138], une approche de génération de trajectoire est proposée en prenant en compte les contraintes mécaniques et cinématiques des robots (*i.e.* la limitation mécanique (physique) de la vitesse, la distance minimale autorisée entre le robot et les obstacles lors de l'évitement d'obstacles, et la limitation mécanique de l'accélération et de la décélération), ainsi que les contraintes dynamiques (*i.e.* la variation maximale de courbure sachant que le rayon de braquage du véhicule dépend de l'angle de braquage des roues avant et arrière, l'erreur causée par l'angle de glissement, et l'adhérence maximale de la roue au sol).

Les auteurs dans [139] s'intéressent particulièrement à la contrainte du temps de navigation tout en prenant en compte les contraintes cinématiques et dynamiques des robots pour générer une trajectoire permettant d'aller du point de départ au point d'arrivée en peu de temps, tandis que dans [140], les auteurs s'intéressent à la notion de robustesse lors de la génération d'une trajectoire pour les robots mobiles.

Dans [141], une approche basée sur le filtrage de la trajectoire en vitesse et en accélération est proposée afin de réduire l'effet des incertitudes et d'avoir une trajectoire réalisable par les robots mobiles à roues.

La notion de minimisation d'énergie est la contrainte principale du travail proposé dans [142], où les auteurs s'intéressent à la génération d'une trajectoire qui permet de consommer le moins d'énergie durant la navigation des robots. Dans [143], les auteurs s'intéressent à une autre contrainte

qui s'impose particulièrement dans les environnements dynamiques, celle du temps de décision des robots mobiles. En effet, les environnements dynamiques imposent une limite supérieure stricte sur le temps disponible pour calculer une trajectoire, car le robot peut être en danger en restant sans mouvement dans un environnement non-fixe. La contrainte du temps de décision est déterminée par la nature des objets en mouvement : plus les objets vont vite, plus le temps de décision et le temps disponible pour calculer une trajectoire est court. Les auteurs dans cette même référence [143] proposent une méthodologie de génération d'une trajectoire en ligne, en prenant en compte cette contrainte du temps de décision plus les contraintes différentielles des robots mobiles à roues.

Les auteurs s'intéressent aux robots bipèdes dans [144] et proposent une méthode de planification de trajectoire utilisant deux types de pendules inversés. La trajectoire se compose de deux phases : Mode pendule inversé linéaire (LIPM), et Mode pendule inversé avec longueur de jambe constante (IPMC). Cela permet au robot de se déplacer comme les humains. Enfin, les auteurs dans [145] s'intéressent à la minimisation du temps de navigation des robots non-holonômes et proposent une approche de génération de trajectoire qui prend en compte les contraintes non-holonômes ainsi que les contraintes physiques (*i.e.* limitation de vitesse et d'accélération) des robots.

I.7 Conclusion

Dans ce chapitre sur l'état de l'art, nous avons présenté une vue d'ensemble sur les différents types des robots mobiles à roues, leur instrumentation, ainsi que des généralités sur les *défauts* et les méthodes de diagnostic. De plus, nous avons présenté un ensemble de travaux liés aux deux problématiques traitées dans cette thèse, à savoir le diagnostic et la navigation autonome des robots mobiles à roues. En effet, ce travail de recherche bibliographique nous a permis d'avoir une idée claire sur le domaine de la robotique mobile à roues.

L'étude des travaux du diagnostic existants dans la littérature nous a permis de bien nous situer par rapport à la littérature, et de choisir les méthodes les plus appropriées à notre cas d'étude en terme d'objectifs. Nous avons utilisé cet état de l'art comme un support pour la synthèse d'une méthodologie permettant le diagnostic des *défauts* les plus probables sur les robots étudiés dans cette thèse, en fonction du modèle choisi et de l'instrumentation embarquée sur ces robots.

De même, l'étude détaillée des approches de navigation nous a permis de comprendre ce qu'il faut maîtriser afin d'avoir un background nous aidant à proposer une méthodologie permettant de compenser le retard généré par les obstacles mobiles imprévus sur le chemin des robots, une problématique qui n'a pas été prise en compte dans les approches de navigation proposées dans la littérature malgré son importance, surtout dans les applications industrielles où le temps d'exécution d'une mission par un robot impacte significativement la production.

CHAPITRE II

DIAGNOSTIC DES *défauts* DES ROBOTS UNICYCLES

II.1	Introduction	37
II.2	Modélisation des robots unicycles	37
II.2.1	Modèle cinématique et dynamique	39
II.2.1.1	Les contraintes non-holonômes	39
II.2.1.2	Modèle cinématique	41
II.2.1.3	Modèle dynamique	42
II.3	Commande des robots unicycles	43
II.4	Diagnostic des <i>défauts</i> capteurs et actionneurs des robots unicycles	44
II.4.1	Mesures disponibles et <i>défauts</i> étudiés	44
II.4.2	Système considéré	46
II.4.3	Principe de l'Observateur Etendu de Kalman (Extended Kalman Observer EKO)	48
II.4.4	Génération des résidus et table de signature	49
II.4.5	Résultats de simulation	53
II.5	Accommodation des <i>défauts</i> capteurs et actionneurs des robots unicycles	57
II.5.1	Résidus et table de signature	59
II.5.2	Estimation des <i>défauts</i> (Fault Estimation FE)	60
II.5.3	Compensation et accommodation des <i>défauts</i>	61
II.5.4	Résultats de simulation	62
II.6	Conclusion	68

II.1 Introduction

Ce chapitre décrit deux travaux que nous avons proposé au cours de cette thèse afin de répondre à la problématique du diagnostic et de l'accommodation des *défauts* du robot unicycle [6, 4]. L'un des objectifs principaux de ces travaux est de proposer une approche simple à implémenter permettant de détecter et d'isoler à la fois les *défauts* des actionneurs et des capteurs embarqués sur le robot unicycle OMRON LD90 auquel nous nous intéressons particulièrement (voir Figure I.5). Cette approche est basée sur la génération d'un ensemble de résidus en utilisant la redondance analytique (*i.e.* des équations analytiques du robot et la théorie de Kalman), combinée avec de la redondance matérielle. Elle nous permet non seulement d'étudier les *défauts* un par un mais elle peut être également étendue pour étudier quelques *défauts* simultanés. Ensuite, une approche basée sur le principe de l'estimation et de la compensation des *défauts* est adaptée afin de permettre au robot d'accomplir sa mission en toute sécurité même en présence d'un *défaut* actionneur ou capteur. Cela vise à éviter l'arrêt immédiat du robot après la détection d'un *défaut* d'une part, et à garantir la stabilité et le contrôle du robot afin d'éviter tout comportement pouvant nuire au robot lui-même et à son environnement d'une autre part. La principale contribution de ces travaux par rapport à la littérature est la considération des *défauts* capteurs et actionneurs à la fois pour ce type de robots. Les *défauts* actionneurs, malgré qu'ils présentent une chance d'apparition assez élevée dû à la sollicitation importante des robots n'ont pas été beaucoup étudiés dans la littérature dans le cas des robots mobiles à roues.

Ce chapitre commence par la modélisation de la cinétique et de la dynamique des robots unicycles dans la Section II.2. Ensuite, afin de pouvoir simuler le comportement du robot unicycle en absence et en présence de *défauts* et tester nos approches, nous utilisons une loi de commande permettant au robot d'aller d'un point initial à un point final ou désiré. La Section II.3 est dédiée à la description de cette loi de commande utilisée dans les deux travaux. Ensuite, l'approche du diagnostic des *défauts* considérés est représentée dans la Section II.4, suivi de quelques résultats de simulation. Enfin, dans la Section II.5, nous détaillons l'approche proposée pour l'accommodation des *défauts* étudiés. Nous présentons également quelques résultats de simulation pour illustrer le comportement du robot en absence de *défauts*, ainsi qu'en présence de *défauts* dans le cas sans compensation, puis avec compensation, afin de montrer l'efficacité de l'approche proposée.

II.2 Modélisation des robots unicycles

La modélisation consiste à mettre au point un ensemble d'équations ou de règles permettant de décrire un phénomène de façon reproductible et simulable. La modélisation est un élément important en robotique. Le modèle issu de l'étape de modélisation sert à synthétiser des lois de commande pour une navigation autonome des robots, à prédire le comportement des robots, et aussi à diagnostiquer les *défauts* qui peuvent apparaître sur différents niveaux des robots.

Souvent, afin de simplifier le problème de navigation des robots mobiles, seul le mouvement du centre de masse G (voir Figure I.3) est pris en compte dans le modèle. Ainsi, le robot est supposé être représenté par un point dans son espace. Ensuite, pour prendre en compte les dimensions du robot, les obstacles statiques sont modélisés dans la carte de l'environnement (map) en les agrandissant par le rayon de la plateforme [146]. Ainsi, les deux problèmes (a) et (b) représentés sur la Figure II.1 deviennent équivalents. Quant aux obstacles mobiles (*i.e.* les obstacles qui ne sont pas schématisés dans la carte de l'environnement car ils ne sont pas statiques, telles que les tables, les chaises, les personnes, ...), ils sont détectés en temps réel par les capteurs extéroceptifs. Comme ces capteurs sont placés aux extrémités du robot, ils estiment la distance réelle entre le robot et les obstacles.

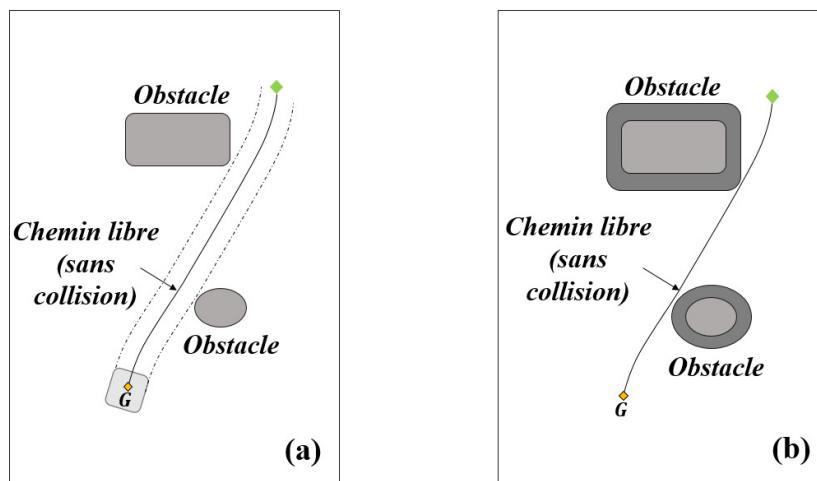


FIGURE II.1 – Deux manières de représenter un robot mobile à roues dans son environnement.

Généralement, on distingue deux façons de modéliser les robots mobiles à roues, selon si l'on considère leur cinétiqe ou leur dynamique. La cinétiqe s'intéresse à la description du mouvement sans considérer les forces qui l'engendrent. L'objectif principal de la modélisation cinématique des robots mobiles à roues est de représenter les vitesses du robot en fonction des vitesses des roues motrices ainsi que des paramètres géométriques du robot. Les modèles cinématiques sont largement utilisés dans la commande des robots mobiles en raison de leur structure simple.

Puisque les modèles cinématiques ne s'intéressent qu'aux vitesses, la dynamique des actionneurs est supposée être assez rapide pour être ignorée, ce qui signifie que les vitesses souhaitées des roues peuvent être atteintes immédiatement [147]. Le modèle dynamique, au contraire, s'intéresse à la fois à la description du mouvement et aux forces qui le provoquent. Il prend en compte le fonctionnement des actionneurs.

Dans cette partie, nous allons décrire les modèles cinématiques et dynamiques des robots unicycles. Nous supposons que les roues sont indéformables, et donc leurs rayons ont une valeur constante. Nous supposons également que le moment d'inertie des robots est invariable.

Le passage du repère local $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ au repère global $(O, \vec{x}, \vec{y}, \vec{z})$ s'effectue via la matrice de transformation $R(\theta)$ de l'équation (II.1), (voir Figure I.3).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R(\theta) \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix}, R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.1})$$

où \dot{x} et \dot{x}_R représentent la vitesse linéaire suivant l'axe \vec{x} dans le repère global et dans le repère du robot respectivement, \dot{y} et \dot{y}_R la vitesse linéaire suivant l'axe \vec{y} dans le repère global et dans le repère du robot respectivement, et $\dot{\theta}$] avec $\dot{\theta}_R$ la vitesse angulaire dans le repère global et dans celui du robot respectivement.

II.2.1 Modèle cinématique et dynamique

Comme mentionné précédemment, nous nous intéressons au robot OMRON LD90 qui est un robot non-holonôme avec deux roues motrices contrôlées séparément, et deux roues folles pour assurer sa stabilité. Pour que ce robot aille de l'avant (suivant une ligne droite), les deux roues motrices doivent avoir la même vitesse et la même direction de rotation, tandis que tourner à gauche (respectivement à droite) est assuré lorsque la roue droite (respectivement gauche) tourne à une vitesse supérieure à celle de la roue gauche (respectivement droite), et finalement pour que le robot tourne autour de lui-même (sur place), les deux roues doivent avoir la même vitesse mais un sens de rotation opposé.

La modélisation cinématique et dynamique des robots unicycles est un sujet qui a été largement abordé dans la littérature, donnant naissance à de nombreuses publications, parmi lesquelles on peut citer [23, 24, 148–159].

Les paramètres géométriques de ce robot unicycle sont illustrés sur le schéma de la Figure II.2, où on a omis les roues folles qui n'interviennent pas dans la cinématique du robot pour plus de clarté sur le schéma. Les variables utilisées pour représenter les paramètres géométriques et le modèle du robot unicycle sont définies dans la TABLE II.1. À noter que les variables $x, y, \theta, \dot{x}, \dot{y}, v, v_r, v_l, \omega, \omega_r, \omega_l, \dot{v}, \dot{\omega}, \tau_r$, et τ_l dépendent du temps t . Pour ne pas alourdir les notations, la variable t n'est pas représentée dans les équations.

II.2.1.1 Les contraintes non-holonômes

La mobilité des robots unicycles sur une surface plane se limite à une translation et une rotation. Ceci s'explique par deux contraintes dites non-holonômes car elles ne sont pas intégrables. Elles sont liées à deux hypothèses issues des conditions du roulement sans glissement suivantes [23] :

(H_1) Pas de glissement latéral ; cette contrainte signifie simplement que le robot ne peut effectuer que des mouvements incurvés (i.e. avant et arrière) mais pas sur le côté. Dans le cadre du robot, cette condition signifie que la vitesse du point G est nulle le long de l'axe \vec{y}_R : $\dot{y}_R^G = 0$ (voir Figure II.2). En utilisant la matrice de passage (II.1), cette contrainte est donnée dans le repère

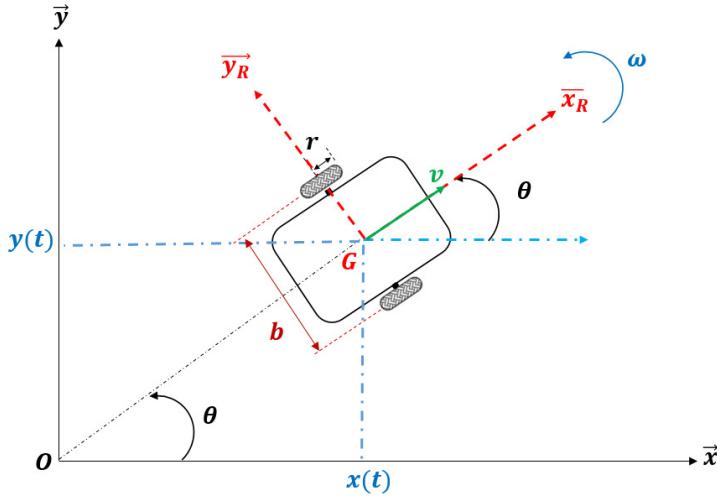


FIGURE II.2 – Géométrie du robot unicycle.

TABLE II.1 – Définition des variables du modèle mathématique du robot unicycle.

Variable	Description & unité
$(x, y), (x_R, y_R)$	La position du robot suivant \vec{x} et \vec{y} dans les repères $(O, \vec{x}, \vec{y}, \vec{z})$ et $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ respectivement [m].
$(\dot{x}, \dot{y}), (\dot{x}_R, \dot{y}_R)$	La vitesse en translation (linéaire) du robot suivant \vec{x} et \vec{y} dans les repères $(O, \vec{x}, \vec{y}, \vec{z})$ et $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ respectivement [m/s].
θ	L'orientation du robot [rad].
v	La vitesse linéaire du robot [m/s].
v_r, v_l	La vitesse linéaire de la roue droite et gauche [m/s].
ω	La vitesse angulaire du robot [rad/s].
ω_r, ω_l	La vitesse angulaire de la roue droite et gauche [rad/s].
\ddot{v}	L'accélération en translation du robot [m/s^2].
$\ddot{\omega}$	L'accélération angulaire du robot [rad/s^2].
r	Le rayon des roues [m].
b	La longueur de l'essieu [m].
M	La masse globale du robot [kg].
M_c	La masse du robot sans la masse des roues et des actionneurs [kg].
I_z	Le moment d'inertie du robot [$kg.m^2$].
I_w	Le moment d'inertie des roues [$kg.m^2$].
τ_r, τ_l	Le couple appliqué aux moteurs droit et gauche respectivement [N.m].

global $(O, \vec{x}, \vec{y}, \vec{z})$ par l'équation Contrainte1, avec \dot{x}_G et \dot{y}_G la vitesse linéaire du point G suivant \vec{x}_R et \vec{y}_R .

$$-\dot{x}^G \sin \theta + \dot{y}^G \cos \theta = 0 \quad (\text{II.2})$$

(H₂) Le contact entre la roue et le sol est ponctuel ; il n'y a pas de patinage ou de glissement de la roue suivant son axe longitudinal (\vec{x}), et il n'y a pas de dérapage suivant son axe orthogonal (\vec{y}). A l'issu de cette condition, la vitesse du point de contact, noté P entre les deux roues (droite et gauche) et le sol (voir Figure I.1, Section I.2.1.1), est supposé égale à $v_i^P = r\omega_i$,

$i \in [r, l]$. Cette contrainte s'exprime pour les deux roues dans le repère global $(O, \vec{x}, \vec{y}, \vec{z})$ en utilisant la matrice de passage (II.1) comme suit :

$$\begin{cases} \dot{x}_r^P \cos\theta + \dot{y}_r^P \sin\theta = r\omega_r \\ \dot{x}_l^P \cos\theta + \dot{y}_l^P \sin\theta = r\omega_l \end{cases} \quad (\text{II.3})$$

avec \dot{x}_i^P et \dot{y}_i^P , $i \in [r, l]$ la vitesse linéaire du point P suivant \vec{x}_R et \vec{y}_R .

II.2.1.2 Modèle cinématique

En tenant compte de l'hypothèse (H_2), les vitesses linéaires des deux roues motrices droite et gauche, v_r et v_l respectivement, sont données en fonction des vitesses angulaires des roues ω_r et ω_l respectivement comme suit [24] :

$$\begin{cases} v_r = r\omega_r \\ v_l = r\omega_l \end{cases} \quad (\text{II.4})$$

La vitesse linéaire v du robot liée au centre de masse G (voir Figure II.2), est donnée en fonction des vitesses angulaires des deux roues motrices ω_r et ω_l comme suit [23], r étant le rayon des roues.

$$v = \frac{v_r + v_l}{2} = r \frac{\omega_r + \omega_l}{2} \quad (\text{II.5})$$

Quant à la vitesse angulaire du robot au point G notée ω , elle est donnée par :

$$\omega = \frac{v_r - v_l}{2} = r \frac{\omega_r - \omega_l}{b} \quad (\text{II.6})$$

En prenant en compte la première contrainte holonôme (II.2), les vitesses du robot (liées au centre de masse G) dans le repère local $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ sont données comme suit :

$$\begin{cases} \dot{x}_R = v \\ \dot{y}_R = 0 \\ \dot{\theta} = \omega \end{cases} \quad (\text{II.7})$$

ce qui s'écrit sous la forme matricielle suivante :

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{II.8})$$

A partir des relations (II.1) et (II.8), le modèle cinématique du robot unicycle est donné dans le repère global $(O, \vec{x}, \vec{y}, \vec{z})$ comme suit :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{II.9})$$

II.2.1.3 Modèle dynamique

Différentes approches ont été proposées pour modéliser la dynamique des robots unicycles, telles que la méthode du Lagrangien [23, 149–155], ou la méthode de Newton-Euler [156–160] ou encore la méthode de Kane [161]. Toutes ces approches permettent d'avoir un modèle mathématique du robot permettant de prendre en compte les forces qui engendrent le mouvement. Dans cette partie, nous présentons le modèle proposé par [23], en utilisant la méthode de Lagrange.

La méthode de Lagrange est une approche très efficace utilisée pour formuler les équations de mouvement des systèmes mécaniques, et ceci en s'appuyant sur leur énergie cinétique et potentielle. L'équation de Lagrange peut s'écrire sous la forme suivante :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) + \frac{\partial L}{\partial q_i} = F - \Lambda^T(q)\lambda \quad (\text{II.10})$$

où $L = T - V$ est la fonction de Lagrange, T et V désignent respectivement l'énergie cinétique et potentielle du système, q_i sont les coordonnées généralisées données dans notre cas d'étude par le vecteur $q = [x, y, \theta, \omega_r, \omega_l]^T$, F est le vecteur des forces généralisées, Λ la matrice des contraintes, et λ est le vecteur des multiplicateurs de Lagrange associés aux contraintes.

Dans le cas du robot unicycle, l'énergie potentielle est nulle $V = 0$ car le robot se déplace sur une surface plane. Quant à son énergie cinétique, elle est donnée par la somme de l'énergie cinétique de la plateforme (sans les roues et les actionneurs) et de l'énergie cinétique des roues et des actionneurs. Elle est exprimée comme suit :

$$T = \frac{1}{2}M(\dot{x}^2 + \dot{y}^2) + M_c d \omega (\dot{y} \cos\theta - \dot{x} \sin\theta) + \frac{1}{2}I_w(\omega_r^2 + \omega_l^2) + \frac{1}{2}I_z \dot{\theta}^2 \quad (\text{II.11})$$

En utilisant la fonction du Lagrange (II.10) avec $L = T$, et en prenant en compte les contraintes non-holonomes (II.2), (II.3) décrites précédemment, le modèle dynamique du robot unicycle est donné par les équations (II.12). Le lecteur peut se référer à [23] pour plus de détails. Ainsi, nous obtenons :

$$\begin{cases} (M + \frac{2I_w}{r^2})\dot{v} - M_c d \omega^2 = \frac{1}{r}(\tau_r + \tau_l) \\ (I_z + \frac{2b^2}{r^2}I_w)\dot{\omega} + M_c d \omega v = \frac{b}{r}(\tau_r - \tau_l) \end{cases} \quad (\text{II.12})$$

où d est la distance entre le centre de masse G et l'origine du repère local. Cette origine est toujours prise au milieu de l'essieu (axe des deux roues motrices). Dans le cas qui nous intéresse ici (*i.e.* robot OMRON LD90), les roues sont placées au milieu du châssis, le centre de masse G coïncide avec l'origine du repère local. De ce fait, $d = 0$.

Comparés aux modèles dynamiques, les modèles cinématiques (dont la commande est en vitesse et non pas en couple) sont beaucoup utilisés dans la commande des robots unicycles suite aux trois raisons suivantes [162] :

1. Le modèle cinématique est plus simple que le modèle dynamique. En particulier, il ne fait pas intervenir un certain nombre de fonctions matricielles dont la connaissance précise repose sur celle des paramètres inertIELS du véhicule et de ses actionneurs (masse, moments d'inertie, coefficients de réduction des moteurs électriques, ...).
2. Pour des robots à motorisation électrique, il est fréquent de disposer d'un asservissement "bas niveau" en vitesse sur les moteurs, dont la grandeur en entrée est une vitesse désirée, et ayant pour rôle d'asservir la vitesse du moteur à celle désirée. Si cet asservissement est bien conçu, l'écart entre ces deux vitesses est toujours faible, de sorte que la vitesse désirée peut à son tour être interprétée comme une variable de commande libre.
3. S'il s'avère que l'asservissement en vitesse de bas niveau et dont le rôle étant de découpler la partie dynamique de la partie cinématique du véhicule n'est pas présent, la façon la plus simple pour la commande étant d'utiliser la commande par "couples calculés" (Computed Torques). Le couple moteur, la vitesse et l'accélération du robot sont liés avec une relation bi-univoque. Cela signifie que le problème de commande en couples se ramène, via la méthode des couples calculés, à un problème de commande en accélération. Il n'est généralement pas difficile de déduire une solution du problème de commande en accélération à partir d'une commande en vitesse.

II.3 Commande des robots unicycles

La problématique de la commande des robots mobiles à roues est trop vaste pour pouvoir être présentée dans ce manuscrit d'une façon exhaustive. Étant donné que ce n'est pas le sujet principal de ce mémoire, nous allons citer quelques références traitant de ce sujet, pour introduire ainsi la loi de commande utilisée dans les travaux de ce chapitre afin de simuler le comportement du robot et tester nos approches.

Les chercheurs se sont intéressés spécialement à deux problèmes liés à la commande des robots mobiles [163], notamment le problème de stabilisation et de suivi de la posture (ou problème de suivi de trajectoire).

Le premier problème vise à trouver une loi de commande permettant de stabiliser la posture (position) des robots non-holonômes à une posture désirée, partant de n'importe quelle position initiale [164, 165]. Quant au second problème, il consiste à commander les robots pour suivre une référence virtuelle ou une trajectoire prédefinie afin d'atteindre la position désirée [163, 166–170].

Dans les travaux de cette thèse, nous ne nous intéressons pas à ces deux problèmes. Le but est d'utiliser une loi de commande qui permet au robot de se déplacer d'un point *A* à un point *B* afin de faire des simulations et tester nos approches. Ainsi, nous optons pour la loi de commande proposée

par le professeur P Corke dans le livre "Robotics, Vision and Control - Fundamental Algorithms in MATLAB" [171]. C'est une loi assez simple qui utilise le modèle cinématique du robot, mais qui permet au robot d'atteindre une position désirée ou une succession de positions désirées dans son environnement d'une façon autonome et efficace.

Considérons le robot unicycle donné par le modèle cinématique décrit par les équations (II.9), où $[x, y, \theta]^T$ constitue le vecteur d'état, et $u = [\nu, \omega]^T$ le vecteur de commande. Partant d'une position quelconque dans son environnement, le robot peut atteindre une position désirée notée $[x_d, y_d]$ en utilisant la loi de commande suivante [171] :

$$\begin{cases} \nu = k_\nu \sqrt{|(x_d - x)^2 - (y_d - y)^2|} \\ \omega = k_\omega \arctan\left(\frac{y_d - y}{x_d - x}\right) \end{cases} \quad (\text{II.13})$$

où k_ν et k_ω désignent deux constantes liées au contrôleur.

II.4 Diagnostic des *défauts* capteurs et actionneurs des robots unicycles

Comme évoqué dans l'**Introduction Générale**, afin de détecter et isoler à la fois les *défauts* capteurs et actionneurs des robots unicycles, nous utilisons dans ce travail une méthode basée sur la combinaison de la redondance analytique (*i.e.* en utilisant des équations analytiques du robot et un banc d'observateurs étendus de Kalman "Extended Kalman Observers" ou EKOs, qui est une des méthodes qui fournit une solution réalisable pour les systèmes stochastiques et non-linéaires [172]) et de la redondance matérielle Afin de pouvoir détecter et parfaitement isoler tous les *défauts* capteurs et actionneurs étudiés.

D'une façon générale, les modèles mathématiques et les observateurs ont les mêmes entrées que le système étudié, avec qui ils fonctionnent en parallèle en temps réel. L'apparition d'un *défaut* conduit à une dissimilarité entre les sorties du système et les sorties estimées par le modèle mathématique ou par les observateurs. Afin d'étudier les *défauts* des capteurs et des actionneurs des robots unicycles, la combinaison de la redondance analytique avec la redondance matérielle est proposé dans ce travail [6] afin de générer un ensemble de résidus ayant des signatures différentes pour chaque *défaut*. Le but est de faciliter l'isolation (*i.e.* la localisation) de l'ensemble des *défauts* étudiés.

Nous utilisons dans ce travail le modèle cinématique des robots unicycles donné par (II.9) dans la Section II.2.1.2. Ce modèle permet d'avoir les sorties dont on dispose dans le robot OMRON LD90, et donc permet d'étudier les *défauts* des capteurs qui nous intéressent.

II.4.1 Mesures disponibles et *défauts* étudiés

Dans le cadre de ce travail, nous disposons de cinq mesures sur le robot unicycle OMRON LD90 :

- La position (x, y) du robot suivant l'axe \vec{x} et \vec{y} , fournie par un système de localisation.

- L'orientation du robot dans son espace de fonctionnement (*i.e.* l'angle θ que forme le robot avec l'axe \vec{x} du repère global $(O, \vec{x}, \vec{y}, \vec{z})$, voir Figure II.2, mesurée à l'aide d'un gyroscope).
- La vitesse angulaire de la roue droite ω_r et de la roue gauche ω_l , mesurée à l'aide de deux codeurs optiques intégrés sur les roues.

Quant aux *défauts* étudiés, nous nous intéressons dans ce travail aux *défauts* des actionneurs et capteurs suivants :

a) *Défauts* actionneurs :

1. *Défauts* de l'actionneur (*moteurs*) de la roue droite noté W_r .
2. *Défauts* de l'actionneur (*moteurs*) de la roue gauche noté W_l .

b) *Défauts* capteurs :

1. *Défauts* du codeur optique droit noté E_r .
2. *Défauts* du codeur optique gauche noté E_l
3. *Défauts* gyroscope noté G .

Quant au type des *défauts* étudiés, nous nous intéressons particulièrement aux *défauts* graduels (*i.e.* dérives), qui sont parmi les *défauts* les plus difficiles à détecter à cause de leur faible amplitude. Ce choix est dû au fait que tous les composants vieillissent avec le temps, même s'ils sont soigneusement conçus, mis en œuvre et testés. De plus, généralement, ces vieillissements se traduisent par des dérives.

Comme la navigation autonome des robots mobiles est basée essentiellement sur le fait d'avoir une position précise, la bonne localisation est une priorité. En effet, comme évoqué précédemment dans la Section I.3.1, la mesure de position (x, y) n'est pas fournie par un seul capteur mais plutôt par différents outils redondants et par combinaison de différentes approches (*e.g.* odométrie, caméras, GPS, ...). Par conséquent, la localisation du robot est supposée être fiable dans ce travail, et les *défauts* du systèmes de positionnement ne sont pas considérés.

De même, les *défauts* système qui se traduisent généralement par le changement des paramètres de la matrice d'état dans le modèle du système ne sont pas considérés pour les deux raisons suivantes :

- D'après les équations du modèle cinématique et dynamique du robot unicycle (II.9), (II.12), les seuls paramètres dont le modèle dépend sont : le rayon des roues r et le moment d'inertie du robot I_z qui sont supposés être invariants (voir Section II.2), la moitié de la longueur de l'essieu b qui est fixe, et la masse du robot M .
- La masse du robot M varie selon si le robot est vide ou chargé, et selon la masse de sa charge (*i.e.* le nombre de FOUPs ainsi que le nombre de wafers dans chaque FOUP transporté par le robot). Ces mesures sont prises en compte par l'entreprise dans le processus de fabrication, et la masse du robot est donc mise à jour à chaque mission en fonction de ces variations.

Le vrai challenge de ce travail est de distinguer les *défauts* codeurs optiques (capteurs) des *défauts* actionneurs, dont l'information de l'état de fonctionnement est disponible via la même source (*i.e.* les codeurs optiques), ainsi que de détecter et d'isoler les *défauts* gyroscope qui impactent considérablement le comportement du robot.

II.4.2 Système considéré

Le modèle cinétique non-linéaire du robot unicycle donné précédemment dans la Section II.2.1.2 par (II.9) peut s'écrire sous forme de la représentation d'état suivante :

$$\begin{cases} \dot{X} = F(X, u) + Bu \\ Y = CX \end{cases} \quad (\text{II.14})$$

avec $X = [x, y, \theta]^T$ le vecteur d'état, $F(X, u)$ est une fonction non-linéaire donnée par $\begin{bmatrix} v \cos \theta \\ v \sin \theta \\ 0 \end{bmatrix}$, B

la matrice de commande donnée par $\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$. Enfin, $Y = X$ représente le vecteur des mesures et C la

matrice d'observation donnée par $I_{(3 \times 3)}$.

Afin de prendre en considération les *défauts* des codeurs optiques E_r et E_l , nous prenons en compte les mesures de la vitesse angulaire des deux roues (droite et gauche) notées respectivement ω_r et ω_l . Ces vitesses sont déduites à partir des équations (II.5) et (II.6) comme suit :

$$\begin{cases} \omega_r = \frac{1}{r}v + \frac{b}{2r}\omega \\ \omega_l = \frac{1}{r}v - \frac{b}{2r}\omega \end{cases} \quad (\text{II.15})$$

Le système (II.14) s'écrit alors sous la forme matricielle suivante :

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}}_{\dot{X}} = \underbrace{\begin{bmatrix} v \cos \theta \\ v \sin \theta \\ 0 \end{bmatrix}}_{F(X, u)} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_B \underbrace{\begin{bmatrix} v \\ \omega \end{bmatrix}}_u \quad (\text{II.16})$$

$$\underbrace{\begin{bmatrix} x \\ y \\ \theta \\ \omega_r \\ \omega_l \end{bmatrix}}_Y = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_C X + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{r} & \frac{b}{2r} \\ \frac{1}{r} & -\frac{b}{2r} \end{bmatrix}}_D \underbrace{\begin{bmatrix} v \\ \omega \end{bmatrix}}_u \quad (\text{II.17})$$

Ce système considéré peut s'écrire sous la représentation d'état suivante :

$$\begin{cases} \dot{X} = F(X, u) + Bu \\ Y = CX + Du \end{cases} \quad (\text{II.18})$$

avec D la matrice d'action directe.

En pratique les mesures sont fournies à des moments définis par un temps d'échantillonnage T_e . Il est préférable de discréteriser le modèle d'état donné par la représentation d'état (II.18). En prenant en compte le vecteur des incertitudes sur le modèle $w(k)$ et le vecteur des bruites de mesure $z(k)$, une telle discritisation peut être donnée approximativement en utilisant la méthode d'Euler comme suit :

$$\begin{cases} X(k+1) = F_d(X, u) + Bu(k) + w(k) \\ Y(k) = CX(k) + Du(k) + z(k) \end{cases} \quad (\text{II.19})$$

où k correspond à l'instant d'échantillonnage, $F_d(X, u) = X(k) + T_e F(X, u)$ la forme discrète de $F(X, u)$. À noter que les vecteurs $w(k)$ et $z(k)$ dépendent de l'instant k . Ils sont supposés être des bruits gaussiens avec des matrices de covariance connues, notée Q et S respectivement et qui sont diagonales. On suppose également que w n'a pas de corrélation croisée (Cross-Correlation) avec z [173]. Les vecteurs $w(k)$ et $z(k)$ ont les propriétés suivantes :

$$\begin{aligned} \mathbb{E}[w(k_1) \ w(k_2)^T] &= \begin{cases} Q & \text{if } k_1 = k_2 \\ 0 & \text{if } k_1 \neq k_2 \end{cases} \\ \mathbb{E}[z(k_1) \ z(k_2)^T] &= \begin{cases} S & \text{if } k_1 = k_2 \\ 0 & \text{if } k_1 \neq k_2 \end{cases} \end{aligned}$$

$$\mathbb{E}[w(k_1) \ z(k_2)^T] = 0, \forall k_1, k_2$$

où $\mathbb{E}[\cdot]$ représente l'espérance mathématique et k_i ($i = 1, 2$) des instants temporels. En pratique, plusieurs méthodes basées sur les données sont utilisées afin de déterminer les matrices Q et S , tel que [174–176]. En simulation, ces matrices sont définies comme une partie liée aux incertitudes du modèle et au comportement du bruit dans le vecteur de mesure. Par conséquent, elles peuvent être déterminées assez facilement.

Observabilité du système (II.18) : Le système (II.18) est observable car tout le vecteur d'état X est mesuré.

Influence du temps d'échantillonage T_e sur l'observabilité du système discret (II.19) :

D'après [177], le système à temps discret (II.19) perd son observabilité s'il existe deux valeurs propres distinctes λ_1 et λ_2 de la matrice $A_l(k)$ (voir (II.21)) qui possèdent la même partie réelle et dont la différence des parties imaginaires est un multiple de la pulsation d'échantillonage $\omega_e = \frac{2\pi}{T_e}$. Dans notre cas, la matrice $A_l(k)$ possède trois valeurs propres identiques $\lambda_1 = \lambda_2 = \lambda_3 = 1$ dont

la partie imaginaire est nulle. Par conséquence, le système (II.18) ne perd pas son observabilité en temps discret. À noter que dans ce cas d'étude, T_e est pris comme étant $T_e = 10^{-4}s$.

II.4.3 Principe de l'Observateur Etendu de Kalman (Extended Kalman Observer EKO)

Le filtre ou l'observateur de Kalman est une méthode de prédiction corrective (Corrective Prediction method) donnée par deux étapes principales [172, 82, 178] : 1) Phase de prédiction et, 2) Phase de correction.

Pour l'étape de prédiction, l'état à l'instant $k + 1$ est défini en connaissant l'instant k par :

$$\begin{cases} \hat{X}(k+1|k) = F_d(\hat{X}(k|k), u(k)) + Bu(k) \\ P(k+1|k) = A_l(k)P(k|k)A_l^T(k) + Q \end{cases} \quad (\text{II.20})$$

où $P(k+1|k)$ désigne la matrice de covariance *a priori* de l'erreur d'estimation associée au vecteur d'état X , et A_l la matrice linéarisée de la fonction $F_d(X, u)$ comme suit :

$$A_l(k) = \frac{\partial F_d(X, u)}{\partial X} \Big|_{\hat{X}(k|k)} = I + T_e \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & & & \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}_{\hat{X}(k|k)} \quad (\text{II.21})$$

Quant à l'étape de correction, elle contient trois équations :

$$\begin{cases} \hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)[Y(k+1) - (C\hat{X}(k+1|k) + Du(k))] \\ K(k+1) = P(k+1|k)C^T(CP(k+1|k)C^T + S)^{-1} \\ P(k+1|k+1) = (I - K(k+1)C)P(k+1|k) \end{cases} \quad (\text{II.22})$$

où I désigne la matrice d'identité avec la dimension appropriée et $K(k+1)$ la matrice de gain du filtre. Pour démarrer les itérations, les conditions initiales sont données par : $\hat{X}(0|0) = \hat{X}_0$, $P(0|0) = P_0$. Les valeurs P_0 sont choisies suffisamment grandes pour couvrir toutes les erreurs d'estimation initiales.

En présence de *défauts* actionneurs et capteurs, le système discret considéré (II.19) s'écrit sous la forme suivante :

$$\begin{cases} X(k+1) = F_d(X, u) + Bu(k) + E_a f_a(k) + w(k) \\ Y(k) = CX(k) + Du(k) + E_s f_s(k) + z(k) \end{cases} \quad (\text{II.23})$$

où E_a et E_s représentent respectivement les matrices d'incidence des *défauts* actionneurs et des

défauts capteurs. Elles sont données respectivement par $E_a = \begin{bmatrix} 0_{3 \times 2} \\ \frac{1}{r} & \frac{b}{2r} \\ \frac{1}{r} & -\frac{b}{2r} \end{bmatrix}$ et $E_s = I_{(5 \times 5)}$ où I est une matrice identité. Enfin, $f_a(k)$ représente le vecteur des *défauts* actionneurs et $f_s(k)$ le vecteur des

défauts capteurs donnés respectivement par $f_a = [W_r, W_l]^T$, $f_s = [0, 0, G, E_r, E_l]^T$. À rappeler que W_i représente le *défaut* actionneur de la roue droite ($i = r$) ou gauche ($i = l$), G le *défaut gyroscope* et E_i le *défaut codeur optique* droit ($i = r$) ou gauche ($i = l$).

Remarque : les éléments W_r, W_l du vecteur f_a et les éléments G, E_r, E_l du vecteur f_s ne sont différents de zéro que si un *défaut* au niveau de ces composants apparaît.

En utilisant l'approximation linéaire A_l de la fonction $F_d(X, u)$ (II.21), l'erreur d'estimation d'état avec l'observateur de Kalman $e(k+1)$ est donnée par :

$$\begin{aligned} e(k+1) &= X(k+1) - \hat{X}(k+1) \\ &\approx (A_l - K(k)C)e(k) + E_a f_a(k) + w(k) - K(k)E_s f_s(k) - K(k)z(k) \end{aligned} \quad (\text{II.24})$$

Dans l'observateur de Kalman, le gain $K(k)$ est calculé de telle sorte à minimiser la trace de la matrice de variance de l'erreur d'estimation (II.25), dans le cas sans *défauts*.

$$\text{Var}(e(k+1)) = (A_l - K(k)C)\text{Var}(e(k))(A_l - K(k)C)^T + \text{Var}(w(k)) + K\text{Var}(v(k))K(k)^T \quad (\text{II.25})$$

avec $\text{Var}(w(k)) = Q$, $\text{Var}(v(k)) = S$, et $\text{Var}(e(k)) = P$.

Par conséquent, lorsqu'un *défaut* actionneur ou capteur apparaît, il est détecté par l'erreur d'estimation d'état (II.24) dont l'expression dépend de ces *défauts*.

II.4.4 Génération des résidus et table de signature

Physiquement, les *défauts* actionneurs W_i et codeurs optiques E_i , $i \in [r, l]$ impactent les vitesses de rotation des roues ω_r et ω_l . Mathématiquement, ces *défauts* peuvent-être écrits comme suit :

$$\begin{cases} \omega_r = \omega_r^* + d_r \\ \omega_l = \omega_l^* + d_l \end{cases}$$

où ω_r^* et ω_l^* sont les vitesses angulaires des roues résultante de l'application de la loi de commande en cas d'absence des *défauts* actionneurs et capteurs. Elles sont calculées à partir de l'entrée du système u en utilisant (II.15). Quant à ω_r et ω_l , elles représentent les vitesses résultantes de l'exécution de la loi de commande par les actionneurs défaillants, ou mesurées par des capteurs défaillants. Enfin, d_r avec d_l représentent les *défauts* de l'actionneur ou du capteur droit et gauche respectivement. Afin d'avoir l'impact de ces *défauts* actionneurs sur le système, la commande appliquée par le système est déduite en partant de ω_r et ω_l (avec *défauts*) en utilisant les relations (II.5) et (II.6).

Afin de pouvoir détecter et isoler les *défauts* capteurs et actionneurs définis précédemment, nous utilisons une approche basée sur la combinaison de la redondance analytique (*i.e.* en utilisant les équations (II.15) et un banc d'observateurs étendus de Kalman) et de la redondance matérielle sur la mesure de l'angle d'orientation θ . La redondance matérielle est utilisée pour les trois raisons suivantes :

1. La position sur x et y dépend de la mesure de θ . Ceci fait que l'effet des *défauts* gyroscope sont en partie compensés par la loi de contrôle qui vise à atteindre une position désirée en fonction de la position en cours. En effet, les faibles *défauts* gyroscope mettent beaucoup de temps avant d'être signalés par la redondance analytique.
2. Les *défauts* actionneurs W_r et W_l impactent tous les états du système, y compris la mesure θ du gyroscope G . En effet, utiliser trois gyroscopes permet de bien isoler les *défauts* gyroscope des *défauts* actionneurs.
3. Les gyroscopes sont des capteurs non encombrants et peu coûteux. Cela rend l'utilisation de trois capteurs au lieu d'un seul une solution facilement réalisable.

Le principe de l'approche du diagnostic proposée est illustré dans la Figure II.3, où $\hat{\omega}_j$, ($j \in [r, l]$) est l'estimation de la vitesse ω_j que l'on s'attend à avoir en appliquant la commande u . Ces estimations, représentées dans le schéma illustré dans la Figure II.3 par 'équations analytiques', sont déduites en utilisant l'équation (II.15) . θ_i , ($i = 1, \dots, 3$) est la mesure du gyroscope i , tandis que $\hat{\theta}_i$ définit l'estimation de la sortie du gyroscope i par l'observateur i .

Construction de l'observateur i , ($i = 1, \dots, 3$) : Les trois observateurs sont construits en respectant les deux étapes (prédition et estimation) données par les expressions (II.20) et (II.22), avec $A_l(k)$ qui est calculée en utilisant l'équation (II.21) et en remplaçant $F_d(X, u)$ par son expression et en considérant $n = \dim(X) = 3$ comme suit :

$$A_l(k) = \frac{\partial F_d(X, u)}{\partial X} \Big|_{\dot{X}(k|k)} = I_{(3 \times 3)} + T_e \begin{bmatrix} 0 & 0 & -v \sin \theta \\ 0 & 0 & v \cos \theta \\ 0 & 0 & 0 \end{bmatrix}_{\dot{X}(k|k)} \quad (\text{II.26})$$

Quant au vecteur de mesures considéré pour chaque observateur, il représente une partie du vecteur des mesures du système Y et il est donné comme suit : $\mathcal{Y}_i = [x, y, \theta_i]^T$, avec θ_i la mesure du gyroscope i ($i = 1, \dots, 3$). Dans ce cas là, la matrice d'observation des observateurs, notée \mathcal{C} , est donnée comme suit : $\mathcal{C} = I_{(3 \times 3)}$. Quant aux matrices Q , S et P_0 , elles sont prises comme étant les mêmes pour les trois observateurs comme suit :

$$Q = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.99 \end{bmatrix}, \quad S = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.09 \end{bmatrix}, \quad P_0 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Après la convergence des observateurs vers les sorties du système, la matrice de gain des observateurs suivante est obtenue :

$$K = \begin{bmatrix} 0.99 & 0 & 0 \\ 0 & 0.99 & 0 \\ 0 & 0 & 0.99 \end{bmatrix}$$

Le banc d'observateurs : Comme illustré sur la figure II.3, le banc d'observateurs est construit comme suit :

1. Observateur 1 : avec cinq entrées, notamment l'entrée du système $u = [v, \omega]^T$, et les trois mesures $\mathcal{Y}_1 = [x, y, \theta_1]^T$ avec θ_1 la mesure du premier gyroscope. Cet observateur fait l'estimation de la variable θ_1 .
2. Observateur 2 : avec cinq entrées, celle du système $u = [v, \omega]^T$, et les trois mesures $\mathcal{Y}_2 = [x, y, \theta_2]^T$, où θ_2 désigne la sortie du deuxième gyroscope. En sortie, il estime la sortie θ_2 .
3. Observateur 3 : avec cinq entrées également, celle du système $u = [v, \omega]^T$, et les trois sorties $\mathcal{Y}_3 = [x, y, \theta_3]^T$ avec θ_3 la mesure du troisième gyroscope. En sortie, il estime la variable θ_3 .

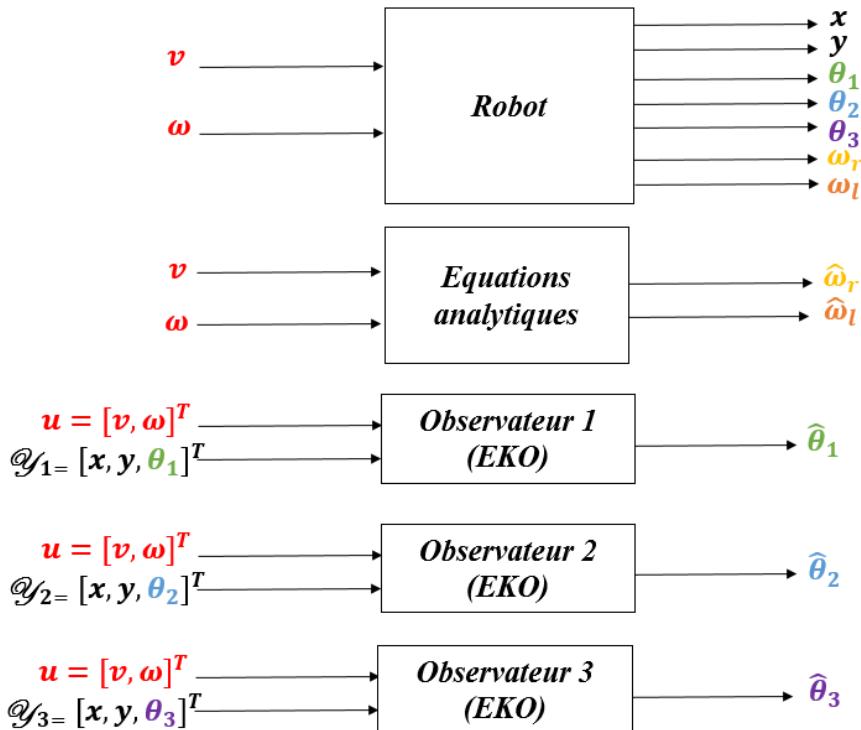


FIGURE II.3 – Principe de détection et d'isolation des *défauts* étudiés.

Avec cette approche, huit résidus sont générés comme suit :

- En utilisant la redondance analytique (*i.e.* le banc d'observateurs), cinq résidus sont générés :

$$\left\{ \begin{array}{l} r_1 = \omega_r - \hat{\omega}_r \\ r_2 = \omega_l - \hat{\omega}_l \\ r_3 = \theta_1 - \hat{\theta}_1 \\ r_4 = \theta_2 - \hat{\theta}_2 \\ r_5 = \theta_3 - \hat{\theta}_3 \end{array} \right. \quad (\text{II.27})$$

- En utilisant la redondance matérielle, trois autres résidus sont générés selon le principe expliqué dans la Section I.4.2.3

$$\begin{cases} r_6 = \theta_1 - \theta_2 \\ r_7 = \theta_1 - \theta_3 \\ r_8 = \theta_2 - \theta_3 \end{cases} \quad (\text{II.28})$$

La signature de ces résidus est déduite en utilisant l'équation (I.1) donnée précédemment dans la Section I.4.2.3, afin de limiter le nombre de fausses alarmes. Les résidus décisionnels R_i , ($i = 1, \dots, 8$) ont les signatures définies dans la Table II.2 en présence des différents *défauts* étudiés. À noter que la notation G_i est utilisée pour désigner le *défaut* du gyroscope i , ($i = 1, \dots, 3$), et W_r , W_l les *défauts* de l'actionneur de la roue droite et gauche respectivement. Enfin, E_r , E_l représentent les *défauts* codeur optique droit et gauche respectivement.

TABLE II.2 – Table de signature des résidus décisionnels en présence des différents *défauts* capteurs et actionneurs étudiés.

	Défaut							
	\emptyset	E_r	E_l	W_r	W_l	G_1	G_2	G_3
Résidu	R_1	0	1	0	1	0	0	0
	R_2	0	0	1	0	1	0	0
	R_3	0	0	0	1	1	1	0
	R_4	0	0	0	1	1	0	1
	R_5	0	0	0	1	1	0	0
	R_6	0	0	0	0	0	1	1
	R_7	0	0	0	0	0	1	0
	R_8	0	0	0	0	0	0	1

Nous pouvons voir grâce à ce tableau que la redondance analytique ne suffit pas pour la détection d'un *défaut* sur le gyroscope. Les autres *défauts* sont détectés et isolés.

Si on s'intéresse au vieillissement des composants, parfois un deuxième *défaut* peut apparaître avant l'isolation ou l'accommodation d'un premier *défaut* détecté, de telle sorte que plus d'un *défaut* soit signalé dans la même période. Dans ce manuscrit, on utilise le terme "*défauts simultanés*" afin de désigner ce cas de figure, sans pour autant dire que les deux *défauts* sont apparus au même instant.

Note : L'apparition de *défauts simultanés* est rare, surtout dans l'industrie où tout est surveillé et où la sécurité est une priorité. Néanmoins, les composants d'un système peuvent vieillir au même temps, et donc le cas des *défauts simultanés* peut arriver. Il est donc très important de pouvoir détecter et surtout d'isoler ces *défauts simultanés*.

Généralement, la détection de plus d'un *défaut* dans un système est facile, mais l'étape d'isolation nécessite plus d'efforts. L'approche que nous proposons dans ce travail permet d'isoler facilement quelques *défauts simultanés*, en se basant juste sur la signature des résidus. La Table II.3 illustre la signature des résidus décisionnels en présence de quelques *défauts simultanés* facilement isolables.

TABLE II.3 – Table de signature des résidus décisionnels en présence de quelques *défauts* simultanés.

		Défaut			
		$E_r + E_l$	$E_r + E_l + G_1$	$W_r + G_1$	$W_l + G_1$
Résidu	R_1	1	1	1	0
	R_2	1	1	0	1
	R_3	0	1	1	1
	R_4	0	0	1	1
	R_5	0	0	1	1
	R_6	0	1	1	1
	R_7	0	1	1	1
	R_8	0	0	0	0

II.4.5 Résultats de simulation

Pour la simulation, les hypothèses suivantes sont prises en compte :

- Au début du fonctionnement du robot, tous ses composants sont supposés être en bon état (*i.e.* sans *défaut*).
- L'amplitude des bruits de mesure est supposée être suffisamment petite devant l'amplitude des *défauts*.
- Les *défauts* sont supposés être permanents et non intermittents.

De plus, les valeurs suivantes sont prises pour les paramètres du robot : $r = 0.095 \text{ m}$ et $b = 0.33 \text{ m}$. En utilisant la méthode des 3σ [89], deux seuils sont fixés pour chaque résidu r_i , ($i = 1, \dots, 8$) sous le mode de fonctionnement nominal du robot. Ces seuils sont appelés $r_{th,i}$ et ils sont donnés pour chaque résidu r_i par :

$$\begin{cases} r_{th,i}^- = -3\sigma_i \\ r_{th,i}^+ = +3\sigma_i \end{cases}$$

où σ_i définit l'écart-type du résidu r_i . La Figure II.4 illustre les seuils définis pour le résidu r_1 .

Comme on peut le remarquer sur la Figure II.4, le fait d'utiliser cette méthode peut générer quelques fausses alarmes. Pour éviter cela, nous utilisons l'algorithme de détection donné précédemment dans la Section I.4.2.3 par l'équation (I.1) suivante :

$$R_i(k) = \begin{cases} 1 & \text{si } |r_i(j)| > r_{th,i}, \forall j \in [k-N+1, k] \\ 0 & \text{sinon} \end{cases}$$

Grâce à cette équation, les résidus décisionnels sont robustes contre les fausses alarmes. A noter qu'ici nous avons pris la valeur de $N = 4$.

Comme évoqué précédemment, on s'intéresse particulièrement aux vieillissement des capteurs et des actionneurs, qui se traduisent généralement par des *défauts* sous forme de dérives (Drift-like

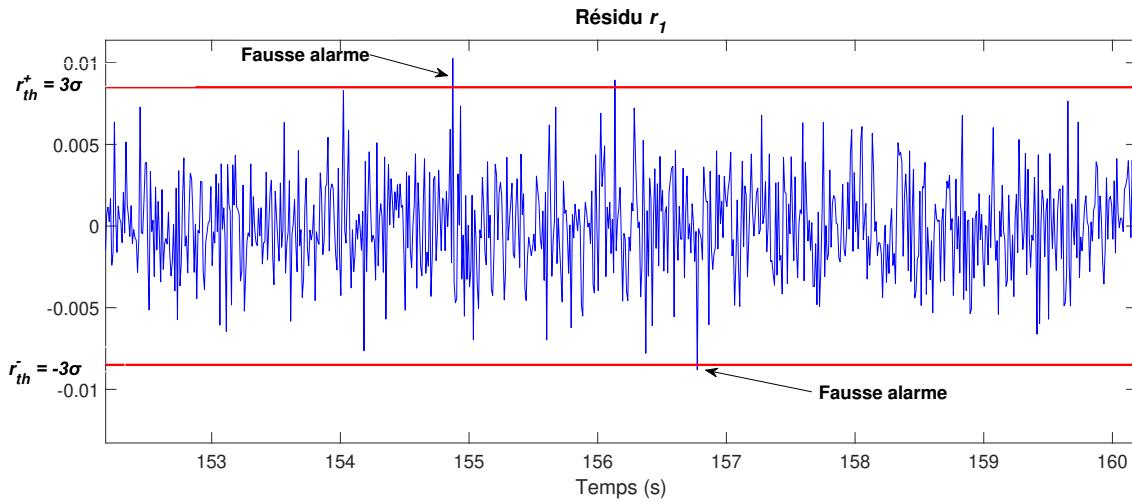


FIGURE II.4 – Définition des seuils $\pm r_{th,1}$.

Faults) à faible amplitude. Afin de simuler ces *défauts*, nous utilisons des signaux sous forme de rampe (Ramp Signals) avec de faibles pentes.

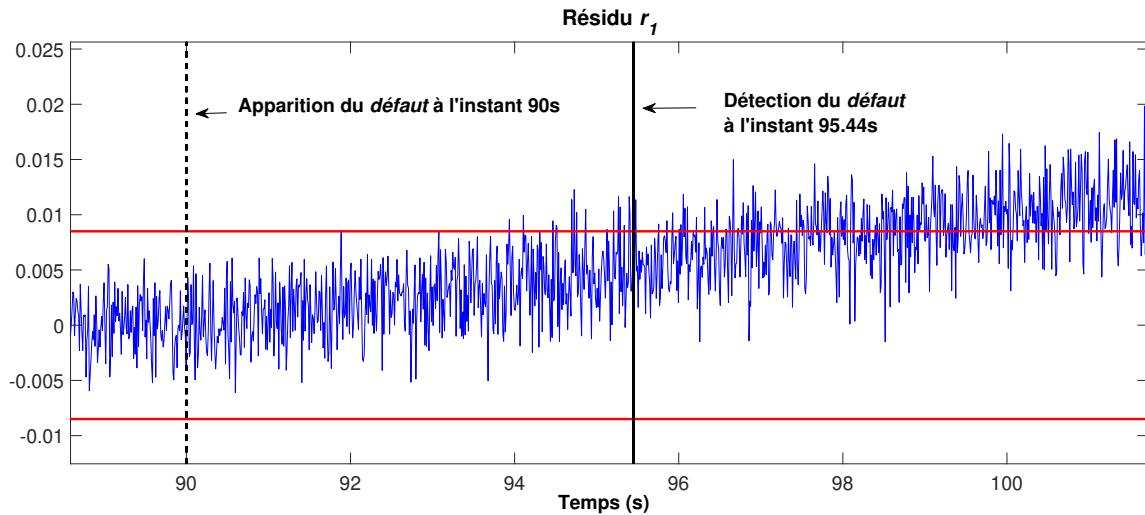
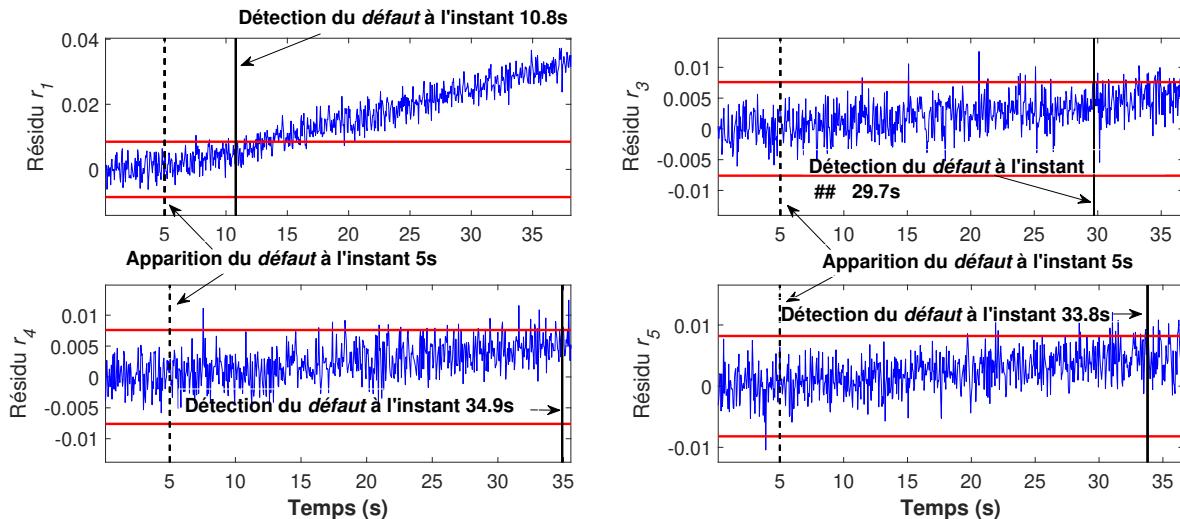
Les *défauts* sont simulés à des instants différents. Les vitesses angulaires ω_r et ω_l des deux roues (droite et gauche) sont mesurées à l'aide des codeurs optiques et elles sont données en rad/s. Connaissant les deux mesures, les vitesses linéaires des deux roues peuvent être déduites en utilisant la relation mathématique (II.4) donnée dans le Section II.2.1.2, avec $r = 0.095\text{ m}$. La vitesse des roues (commande) dépend du chemin de référence qui permet au robot d'aller d'une position initiale à une position désirée (voir les équations (II.13)). Pour le scénario simulé dans ce travail, la vitesse angulaire moyenne déduite pour les deux roues est d'environ 0.02 rad/s , tandis que la valeur moyenne de l'angle θ , mesurée par les gyroscopes G_i est d'environ 2 rad . Afin d'avoir une idée sur l'amplitude des *défauts* simulés, nous les exprimons en pourcentage par rapport à ces valeurs moyennes.

Dans les figures qui suivent, nous illustrons uniquement les résidus qui s'éloignent de zéro, signalant ainsi la présence d'un *défaut*. Le reste des résidus étant proche de zéro comme sur la Figure II.4, ils ne sont donc pas représentés pour des soucis de clarté sur les figures.

Dans la Figure II.5, un *défaut* graduel avec une pente de 10^{-3} rad/s est simulé au niveau du codeur optique de la roue droite E_r . Ce *défaut* représente 5% de la valeur moyenne de la vitesse angulaire ω_r . En vitesse linéaire, c'est l'équivalent d'une déviation de 0.34 m/h de la mesure du codeur E_r . Le *défaut* apparaît à l'instant $t = 90\text{ s}$, il est détecté et isolé juste 5.44 s plus tard.

La Figure II.6 montre la détection et l'isolation d'un *défaut* simulé sur l'actionneur de la roue droite W_r . La dérive simulée a une pente de -10^{-3} rad/s , qui est l'équivalent d'une déviation de $9.5 \cdot 10^{-5}\text{ m/s}$. Ce *défaut* représente 5% de la valeur moyenne de ω_r . Il est détecté 5.8 s après son apparition (voir r_1 Figure II.6), puis isolé 24.1 s plus tard.

La Figure II.7 montre les résultats de simulation d'un *défaut* simulé au niveau du gyroscope G_1 . Ce *défaut* a comme pente la valeur de 10^{-3} rad/s , qui est l'équivalent de 0.0573° , et qui représente 0.05% de la valeur moyenne de θ . Le *défaut* est simulé à l'instant $t = 10\text{ s}$ et il est détecté 14.5 s

FIGURE II.5 – Résidu r_1 en présence d'un défaut sur le codeur optique E_r .FIGURE II.6 – Résidus r_1 , r_3 , r_4 , et r_5 en présence d'un défaut sur l'actionneur W_r .

après son apparition, puis isolé 0.5 s plus tard. Ici le résidu r_3 n'est pas représenté car le défaut simulé est assez faible pour être détecté par r_3 (redondance analytique) sur la durée de simulation.

Dans la Figure II.8, deux défauts sont supposés apparaître dans la même période :

1. Un défaut au niveau de l'actionneur gauche W_l avec une pente de -10^{-3} rad/s , l'équivalent de $-9.5 \cdot 10^{-5} \text{ m/s}$, et qui représente 5% de la valeur moyenne de la vitesse angulaire ω_l .
2. Un défaut au niveau du gyroscope G_2 avec une pente de 10^{-2} rad , l'équivalent de 0.57° , qui représente 0.5% de la valeur moyenne de θ .

Les deux défauts sont simulés et détectés puis isolés respectivement comme suit : le défaut actionneur est apparu à $t = 10 \text{ s}$, puis détecté à 17.4 s et isolé 20.2 s plus tard. Quant au défaut gyroscope G_2 , il est apparu à $t = 12 \text{ s}$, 2 s après l'apparition du défaut actionneur W_l . Il est détecté juste 1.6 s après son apparition et puis isolé 2 s plus tard.

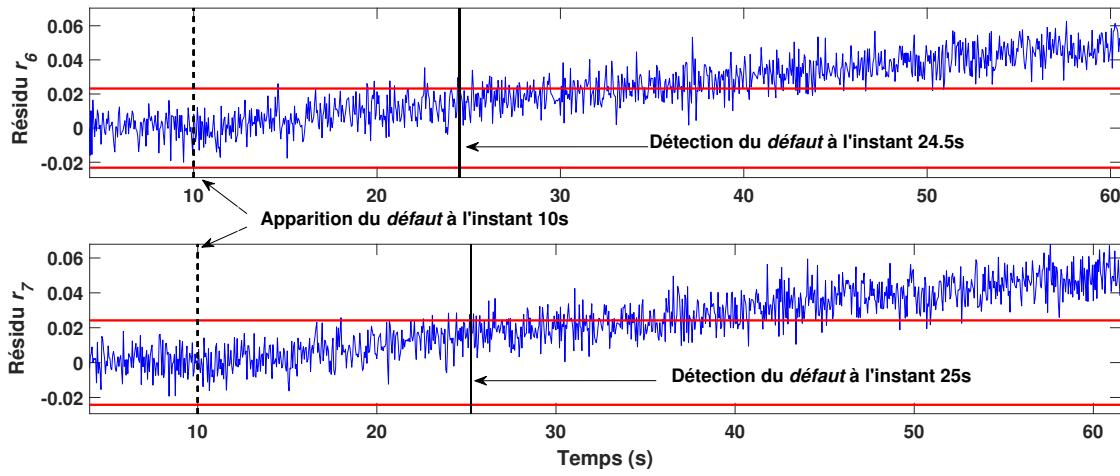


FIGURE II.7 – Résidus r_6 et r_7 en présence d'un défaut gyroscope G_1 .

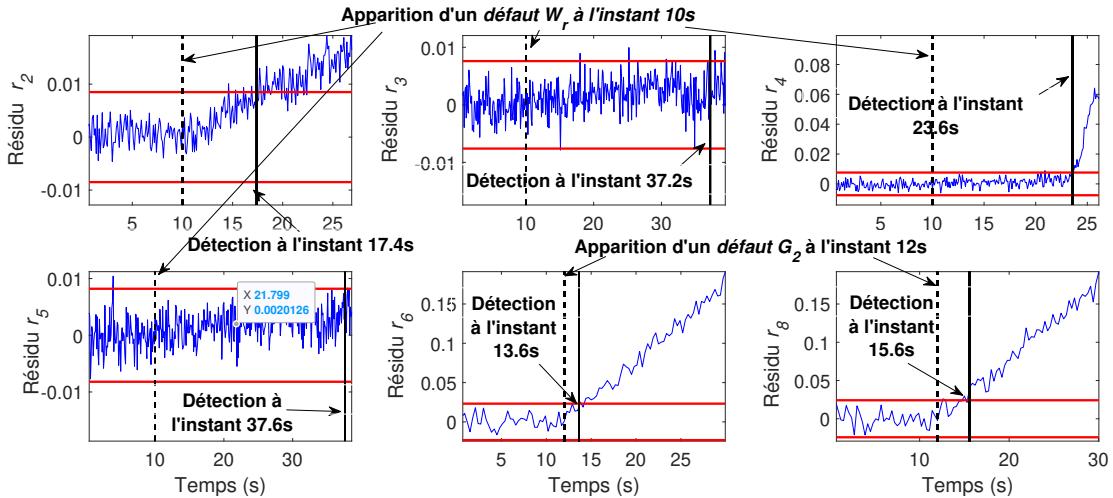


FIGURE II.8 – Résidus r_2 , r_3 , r_4 , r_5 , r_6 , et r_8 en présence d'un défaut actionneur W_l et d'un défaut gyroscope G_2 .

Enfin, la Figure II.9 représente la détection et l'isolation de trois défauts simultanés :

- 1 & 2. Un défaut au niveau des deux codeurs droit et gauche (E_r et E_l) avec une pente de 10^{-3} rad/s $= 9.5 \cdot 10^{-5} \text{ m/s} = 5\%$ de la valeur moyenne de ω_j ($j \in [r, l]$)
3. Un défaut au niveau du gyroscope G_3 avec une pente de $10^{-4} \text{ rad} = 0.0057^\circ = 0.005\%$ de la valeur moyenne de θ .

Les défauts codeurs E_r et E_l sont détectés et isolés en moins de 7 s, tandis que le défaut gyroscope G_3 est détecté 90.3 s après son apparition et puis isolé après 27.7 s supplémentaire. À noter que la vitesse de détection et d'isolation des défauts dépend de l'amplitude du défaut. Plus l'amplitude du défaut est faible, plus il met du temps pour être détecté et isolé. Ici le résidu r_5 (impacté par le défaut G_3) n'est pas représenté car le défaut gyroscope simulé est faible pour être détecté sur l'intervalle de simulation (il est en partie compensé par la loi de contrôle).

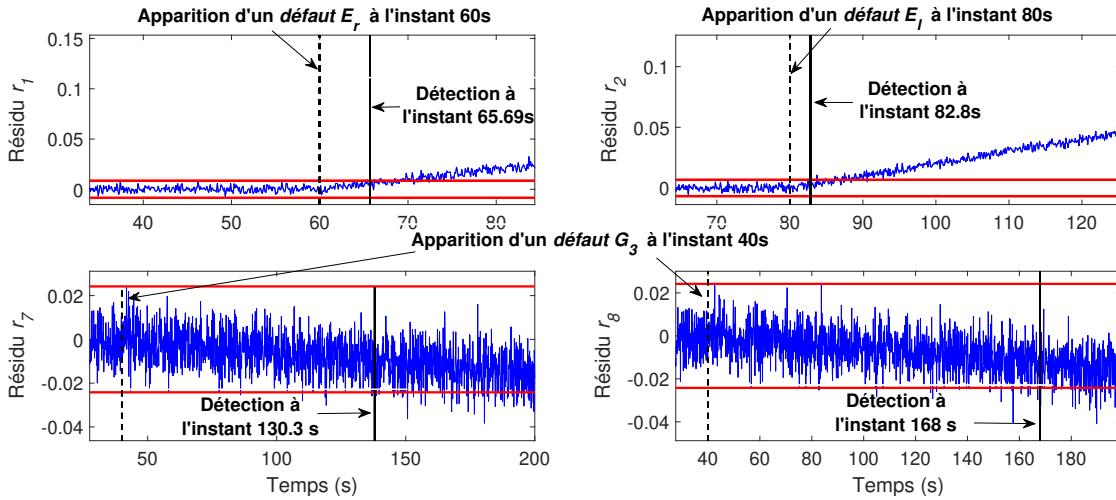


FIGURE II.9 – Résidus r_1 , r_2 , r_7 , et r_8 en présence de *défauts* codeurs droit et gauche, et d'un *défaut* gyroscope G_3 .

II.5 Accommodation des *défauts* capteurs et actionneurs des robots unicycles

Comme mentionné tout au long de la partie précédente de ce **Chapitre II**, la détection et l'isolation des *défauts* internes des systèmes embarqués tels que les robots mobiles à roues, est une étape très importante, et plus cette opération est rapide mieux c'est. Mais quel est l'intérêt de détecter et d'isoler un *défaut*? Pourquoi la rapidité de détection est-elle aussi importante?

Les robots dans la fab évoluent dans un environnement dynamique qui contient des équipements et des opérateurs humains. Le moindre *défaut* peut entraîner des dommages potentiellement importants pouvant nuire à la santé des opérateurs humains ou à l'environnement de fonctionnement du robot. Le fait de détecter et d'isoler un *défaut* très tôt permet d'éviter la défaillance totale du composant en *défaut* d'une part, et d'éviter que le *défaut* impacte le comportement du robot et entraîne des dommages d'une autre part. Mais comment peut-on éviter les conséquences indésirables d'un *défaut* après l'avoir détecté?

Généralement, deux choix sont possibles :

1. Une maintenance corrective, avec un arrêt immédiat du robot et une intervention humaine pour réparer ou changer le composant en *défaut*.
2. La prise en compte du *défaut* dans la boucle du contrôle et la synthèse d'une loi de commande tolérante aux fautes (Fault Tolerant Control ou FTC).

La commande tolérante aux fautes (ou FTC) est l'une des techniques qui permet de maintenir un comportement normal d'un système même en présence d'un *défaut*, sans avoir à l'arrêter immédiatement. Zhang et Jiang définissent les systèmes de commande tolérante aux fautes (Fault Tolerant Control Systems ou FTCS) dans [179] comme étant des systèmes capables de gérer automatiquement leurs défauts internes. Ils sont capables de maintenir leurs performances acceptables en cas de

défaut. Une revue bibliographique non-exhaustive des systèmes de commande tolérante aux fautes reconfigurable peut être trouvée dans cette même référence [179].

Dans la littérature [179, 180], les FTCS sont classés en deux groupes principaux : 1) Systèmes *passifs*, et 2) Systèmes *actifs* (Passive & Active FTCs ou PFTCs & AFTCs). Dans [181], une étude comparative entre les approches PFTC et AFTC est établie. La classification des méthodes FTC est proposée dans [182] (voir Figure II.10).

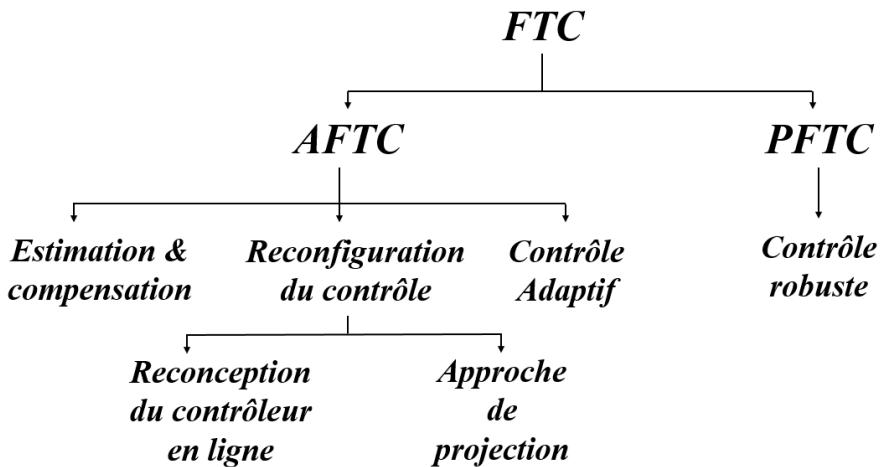


FIGURE II.10 – Classification des approches FTC.

Les approches PFTC sont basées sur le listing des *défauts* potentiels, supposés être connus *a priori* sous le nom de défauts de conception. Ainsi, tous les modes de fonctionnement nominal et en *défaut* sont pris en compte lors de la conception de la loi de commande (contrôle robuste). Ces approches ne nécessitent ni un schéma FDD (approche de détection, d'isolation, et d'estimation du *défaut*) ni un mécanisme de reconfiguration du contrôleur [181]. Ces approches peuvent-être utilisées si la fenêtre de temps pendant laquelle le système reste stabilisable en présence d'un *défaut* est courte [182].

Quant aux approches AFTC, elles nécessitent un schéma FDD, un contrôleur reconfigurable, et un mécanisme de reconfiguration du contrôleur.

- Pour la reconfiguration du contrôle, on distingue généralement deux types d'approches :
 1. Approches basées sur la projection (Projection-based approaches) : une nouvelle loi de commande pré-calculée est sélectionnée en fonction du *défaut* survenu.
 2. Approches de reconfiguration en ligne du contrôleur (Online automatic control redesign approaches) : cela implique le calcul des nouveaux paramètres du contrôleur (contrôle reconfigurable).
- Les méthodes traditionnelles du contrôle adaptatif qui adaptent les paramètres du contrôleur aux modifications du système peuvent être utilisées comme des approches AFTC, mais en supprimant potentiellement la nécessité des étapes FDI/FDD et la reconfiguration du contrôleur [183] et

en utilisant éventuellement l'estimation du *défaut* (Fault estimation ou FE). Ces approches conviennent aux installations dont les paramètres varient lentement et peuvent tolérer les *défaux* actionneurs et système [182].

- Quant à l'approche basée sur l'estimation et la compensation des *défaux*, connue également sous le nom de stratégie de masquage des *défaux*, elle est centrée sur la réalisation des objectifs de boucle fermée telle que la boucle de commande nominale reste inchangée grâce à l'utilisation d'actionneurs virtuels ou de capteurs virtuels [184, 185]. Le masquage des *défaux* utilise la différence entre l'état nominal et l'état défectueux du système pour modifier la dynamique du système de telle sorte que les objectifs requis de la commande sont atteints en permanence même en cas de *défaut*. Dans le cas d'un *défaut* capteur, l'effet du *défaut* est masqué à l'entrée du contrôleur. Cependant, les *défaux* actionneurs sont compensés par l'effet du *défaut* [186, 185]. Cette aproche est efficace à condition qu'un schéma FDD robuste soit disponible.

Dans ce travail [4], nous nous intéressons particulièrement à l'approche basée sur l'estimation et de la compensation des *défaux*. Ce travail est la suite du travail précédent [6] qui propose un schéma pour la détection et l'isolation des *défaux* actionneurs et capteurs embarqués sur les robots unicycles intégrés dans la fab. Il traite le problème de l'accommodation des *défaux* actionneurs (*i.e.* moteur droit et gauche W_r et W_l) et des *défaux* capteurs (*i.e.* codeurs optiques placés sur la roue droite et gauche, notés E_r et E_l respectivement). Nous avons choisi cette approche car elle s'adapte facilement à nos besoins, et elle ne dépend pas de la loi de commande utilisée. Cette approche reste valable dans le cas où une autre loi de contrôle (pour le suivi de trajectoire par exemple) est utilisée. De plus, elle permet de compenser parfaitement les *défaux* considérés comme nous allons le voir dans la Section II.5.4.

Les *défaux* gyroscope ne sont pas considérés ici car comme nous l'avons vu dans la Section II.4 précédente, ils sont détectés en utilisant la redondance matérielle. Cela permet de basculer la mesure vers un gyroscope sans *défaut* (Healthy Gyroscope) après détection d'un *défaut* sur le gyroscope pris en compte au début.

II.5.1 Résidus et table de signature

Parmi les huit résidus générés dans le travail précédent [6], seuls trois nous intéressent dans ce travail [4] car nous ne considérons pas les *défaux* gyroscope et les *défaux* simultanés. Le but ici est d'isoler les *défaux* codeurs optiques et actionneurs, puis de les accommoder (*i.e.* de les compenser). Nous ne considérons qu'un seul observateur étendu de Kalman en s'appuyant sur le modèle du système augmenté (II.16) et les équations analytiques données par (II.15) afin de générer les trois résidus donnés par (II.29), comme illustré sur la Figure II.11.

$$\begin{cases} r_1 = \omega_r - \hat{\omega}_r \\ r_2 = \omega_l - \hat{\omega}_l \\ r_3 = \theta - \hat{\theta} \end{cases} \quad (\text{II.29})$$

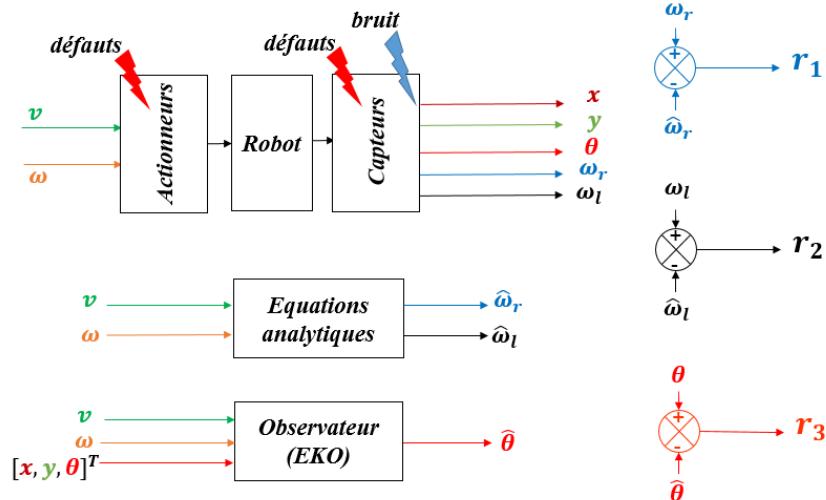


FIGURE II.11 – Génération de trois résidus.

La Table II.4 de signature de ces trois résidus en présence des *défauts* capteurs et actionneurs considérés est définie en utilisant l'équation (I.1) donnée dans la Section I.4.2.3.

TABLE II.4 – La signature des trois résidus en présence des *défauts* codeurs et actionneurs.

		Défaut				
		\emptyset	E_r	E_l	W_r	W_l
Résidu	R_1	0	1	0	1	0
	R_2	0	0	1	0	1
	R_3	0	0	0	1	1

Comme nous pouvons le remarquer dans cette Table II.4, les *défauts* étudiés sont totalement détectés et isolés à l'aide de ces 3 résidus.

II.5.2 Estimation des *défauts* (Fault Estimation FE)

Afin de réduire les fausses alarmes et d'améliorer le temps de détection des *défauts* étudiés, les résidus r_i , ($i = 1, \dots, 3$) sont filtrés du bruit de mesures comme nous l'avons décrit précédemment dans l'étape 1 du traitement de fausses alarmes, Section I.4.2.3. Pour ce faire, un filtre de Kalman (Kalman Filter KF) est utilisé pour le système défini comme suit : (voir Figure II.12)

$$\begin{cases} X_r(k+1) &= A_r X_r(k) + w_r \\ Y_r(k) &= C_r X_r(k) + z_r \end{cases} \quad (\text{II.30})$$

avec X_r le vecteur d'état défini par $X_r = [r_1, r_2, r_3]^T$, $Y_r = X_r$ le vecteur de sortie, A_r et C_r sont des matrices d'identité (3×3). w_r et z_r désignent les incertitudes de modélisation et les bruits de mesure respectivement. Ils ont les mêmes propriétés (bruits blancs, Gaussiens, non-correlés, ...) que w et z du système, et ont comme matrices de covariance respectivement Q_r et S_r . Les valeurs de la matrice S_r correspondent aux variances des résidus en mode nominal, tandis que les valeurs Q_r doivent respecter

le compromis entre le filtrage et le délai ; de très petites valeurs entraînent un retard considérable sur la réponse du filtre, tandis que de grandes valeurs ne permettent pas un bon filtrage.

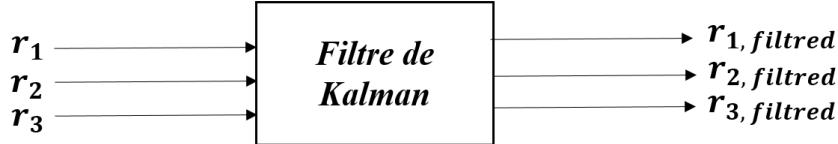


FIGURE II.12 – Principe du filtrage des résidus.

D'après (II.29), les résidus r_1 et r_2 sont associés aux *défauts* sur ω_r et ω_l respectivement. Un *défaut* sur la mesure ω_r (respectivement ω_l) représente un *défaut* sur l'actionneur W_r ou sur le capteur E_r de la roue droite (respectivement sur l'actionneur W_l ou sur le capteur E_l de la roue gauche). Les résidus filtrés $r_{1,filtred}$ et $r_{2,filtred}$ sont donc une image des *défauts* actionneur et codeur droit et gauche respectivement. Par conséquent, ces deux résidus filtrés sont utilisés afin d'estimer les allures de ces *défauts*, et pour ensuite les compenser.

II.5.3 Compensation et accommodation des *défauts*

Dans ce travail, le robot est commandé pour se déplacer d'un point $A = [x, y]^T$ à un point $B = [x_d, y_d]^T$ en utilisant la loi de contrôle donnée précédemment par (II.13) dans la Section II.3. Le but de ce travail [4] consiste à compenser les *défauts* actionneurs et capteurs des deux roues droite et gauche, à savoir les *défauts* des codeurs optiques et des moteurs, et cela après les avoir détectés et bien isolés. Le fait de filtrer les résidus permet non seulement d'améliorer le temps de détection, mais aussi d'estimer les *défauts* via les deux résidus $r_{1,filtred}$ et $r_{2,filtred}$. Afin d'accommoder ces *défauts*, le principe suivant est utilisé : (voir Figure II.13)

- Si un *défaut* apparaît au niveau du codeur E_r (respectivement E_l), c'est-à-dire si $R_1 = 1$ (ou $R_2 = 1$) et $R_3 = 0$ (voir Table II.4), le *défaut* est compensé à la sortie du codeur optique de la roue droite ω_r (respectivement gauche ω_l), de telle sorte à avoir les bonnes mesures ω_r^* (respectivement ω_l^*) malgré la présence du *défaut*, comme suit :

$$\begin{cases} \omega_r^* = \omega_r - r_{1,filtred} \\ \omega_l^* = \omega_l - r_{2,filtred} \end{cases} \quad (\text{II.31})$$

- Si un *défaut* se produit au niveau de l'actionneur de la roue droite W_r (respectivement gauche W_l), c'est-à-dire si $R_1 = 1$ (ou $R_2 = 1$) et $R_3 = 1$ (voir Table II.4), la position du robot x et y est influencée (*i.e.* le robot s'éloigne de la position désirée). Les entrées de commande résultantes de la position défective du robot (voir (II.13)) sont donc incorrectes. Pour tenir compte des *défauts* actionneurs, la nouvelle entrée de commande $[v^*, \omega^*]^T$ est calculée grâce aux équations

(II.5) et (II.6), en compensant les *défauts* estimés comme suit :

$$\begin{cases} v^* = v - \frac{r}{2}(r_{1,filtered} + r_{2,filtered}) \\ \omega^* = \omega - \frac{r}{b}(r_{1,filtered} - r_{2,filtered}) \end{cases} \quad (\text{II.32})$$

où v et ω représentent les entrées appliquées par les actionneurs en *défaut*.

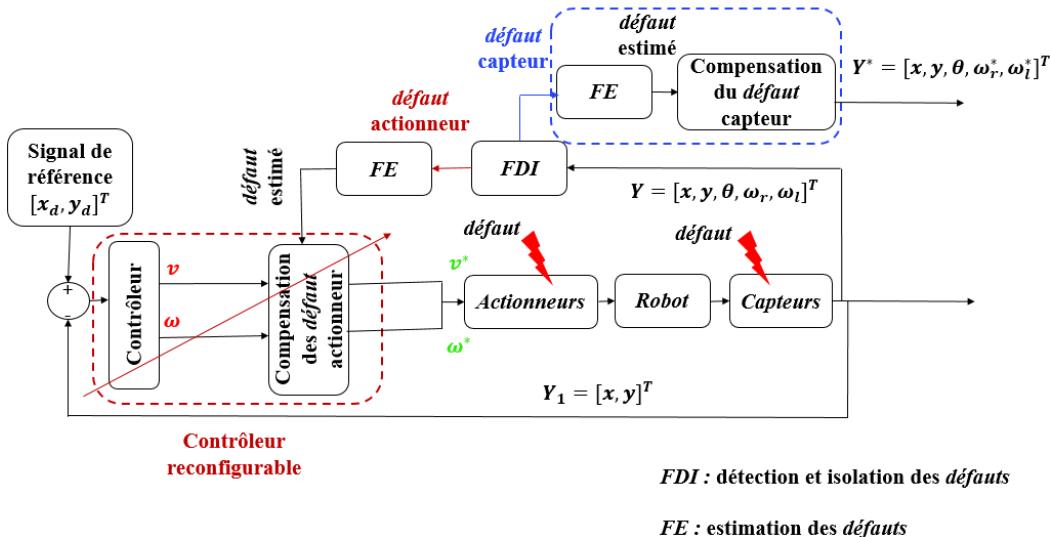


FIGURE II.13 – Principe de compensation des *défauts* capteurs et actionneurs des roues.

II.5.4 Résultats de simulation

Les mêmes valeurs que le travail précédent des paramètres r et b sont prises : $r = 0.095\ m$ et $b = 0.33\ m$. Dans ce qui suit, juste les résidus sensibles aux *défauts* simulés (*i.e.* les résidus qui s'éloignent de zéro) sont illustrés. Les autres n'étant pas sensibles, ils restent dans l'intervalle $\pm 3\sigma$ et ils ne sont donc pas présentés.

Afin de montrer les avantages du filtrage de bruit de mesures sur la rapidité de détection, la Figure II.14 montre le résidu r_1 avant et après le filtrage du bruit sur la mesure du codeur optique E_r , en présence d'un *défaut* sur ce capteur avec une pente de $10^{-2}\ rad/s$.

Avant le filtrage (*i.e.* avec $r_{1,non,filtered}$), le *défaut* met 0.53 s afin d'être détecté et isolé, tandis qu'il ne met plus de 0.19 s avec $r_{1,filtered}$, (*i.e.* après le filtrage de r_1). Dans les résultats qui suivent, nous ne présentons que le diagnostic des *défauts* simulés avec les résidus filtrés $r_{i,filtered}$, ($i = 1, \dots, 3$).

La Figure II.15 montre la détection et l'isolation d'un *défaut* simulé au niveau de l'actionneur de la roue gauche W_l . Le *défaut* a comme pente la valeur de $-10^{-2}\ rad/s$, ce qui représente un

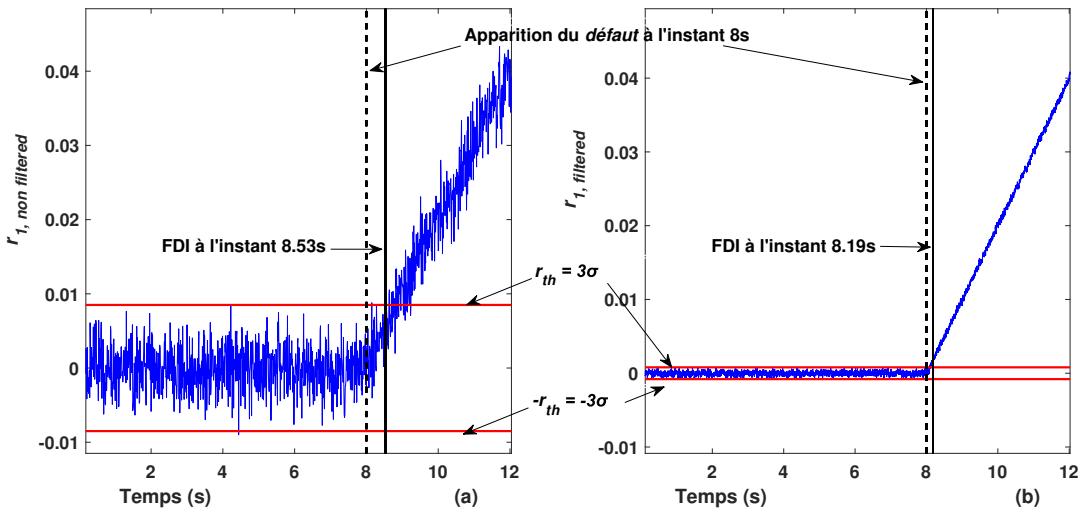


FIGURE II.14 – Détection et isolation d'un *défaut* au niveau du codeur de la roue droite avec le résidu r_1 avant (a) et après (b) filtrage.

vieillissement de l'ordre de 3.6 rad/h . Avec les résidus filtrés, le *défaut* a mis juste 0.04 s pour être détecté avec $r_{3,filtered}$, et juste 0.06 s plus tard, il a été isolé par $r_{2,filtered}$, voir la Figure II.15.

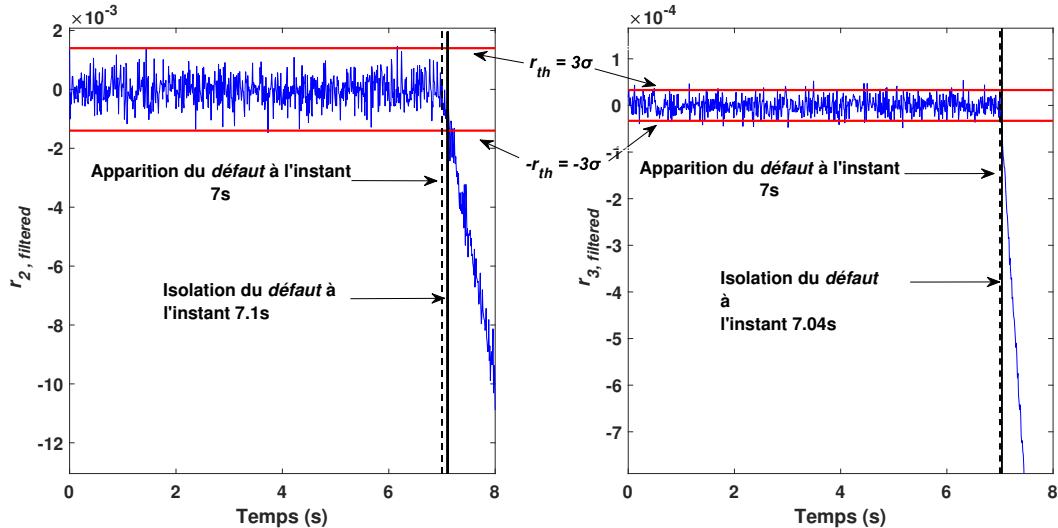


FIGURE II.15 – Détection et isolation d'un *défaut* au niveau de l'actionneur de la roue gauche.

Afin d'illustrer le principe d'estimation des *défauts*, le *défaut* simulé sur l'actionneur de la roue gauche W_l est schématisé sur la même Figure II.16 avec le résidu filtré $r_{2,filtered}$. Comme on peut le voir sur cette figure, le *défaut* est parfaitement reconstruit avec $r_{2,filtered}$.

Le résultat de la compensation des *défauts* capteurs est illustré sur la Figure II.17, où un *défaut* avec une pente de 10^{-1} rad/s est simulé au niveau du codeur optique de la roue gauche E_l . La vraie vitesse angulaire de la roue (*i.e.* sans le *défaut* capteur) est donnée en vert, tandis que la mesure du codeur après le *défaut* et sans compensation est donnée en rouge, et cette même mesure après la

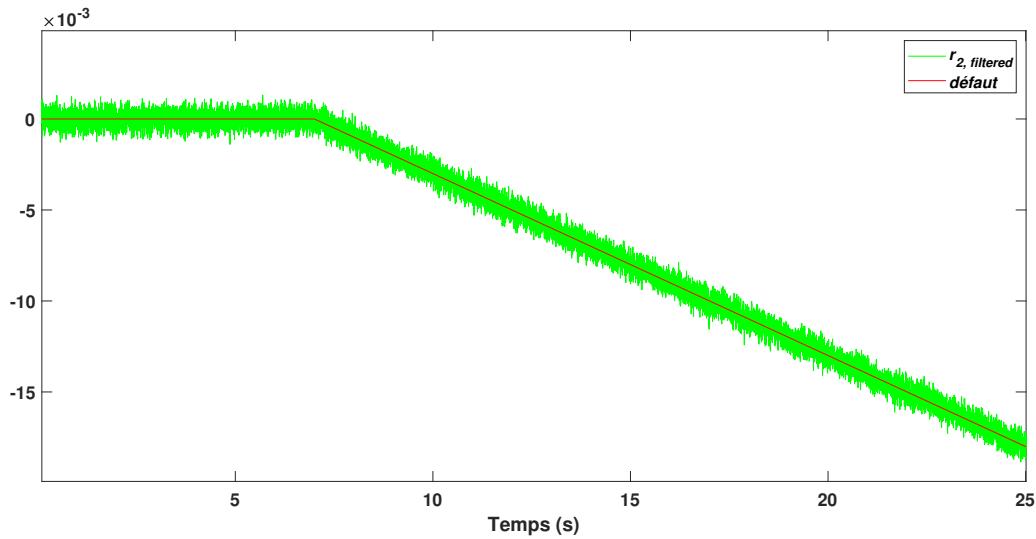


FIGURE II.16 – Estimation du *défault* simulé sur W_l avec $r_{2,filtered}$.

compensation du *défault* est donnée en bleu. Comme on peut le constater sur ce résultat, le codeur optique envoie la vraie mesure même en présence du *défault* grâce au principe de compensation appliqué (*i.e.* le *défault* est parfaitement compensé).

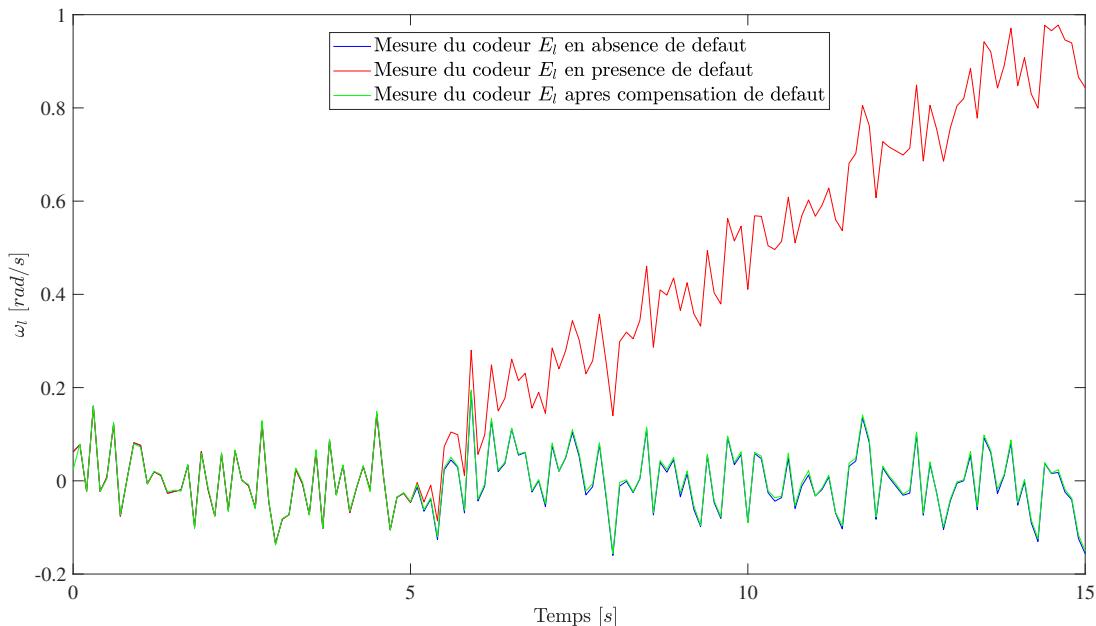


FIGURE II.17 – Compensation d'un *défault* simulé au niveau du codeur E_l .

Afin d'illustrer l'efficacité de la compensation des *défaux* actionneurs, les figures II.18 et II.19 montrent respectivement le chemin de déplacement du robot unicycle entre différents points (Robot path) de son environnement selon l'axe des x et des y et sur le plan $y(x)$, comparés au chemin désiré (Desired path) en utilisant la loi de commande (II.13) et en absence de *défaux* capteurs et actionneurs.

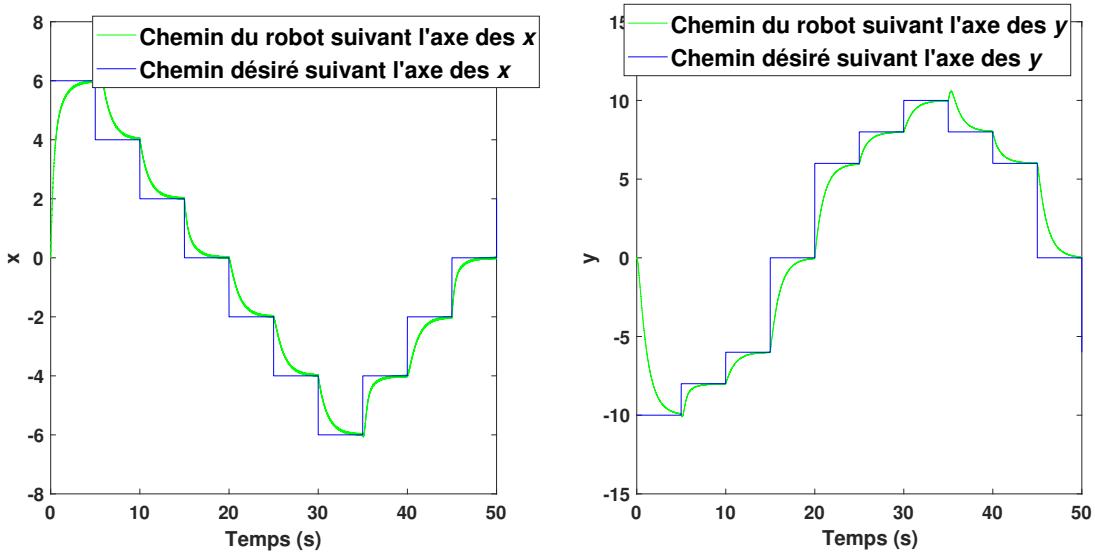


FIGURE II.18 – Chemin de déplacement du robot unicycle selon l'axe x et y comparés aux chemins désirés, en absence de défauts.

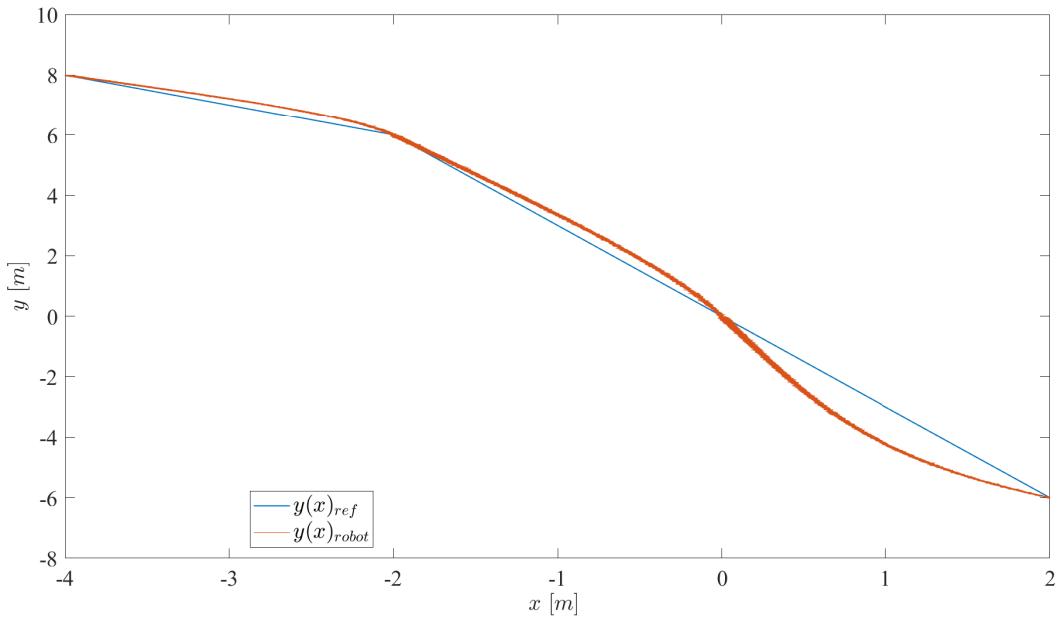
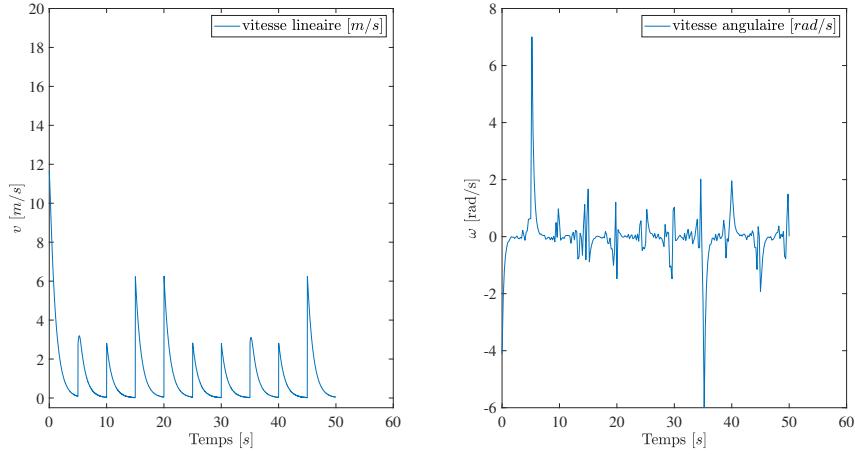
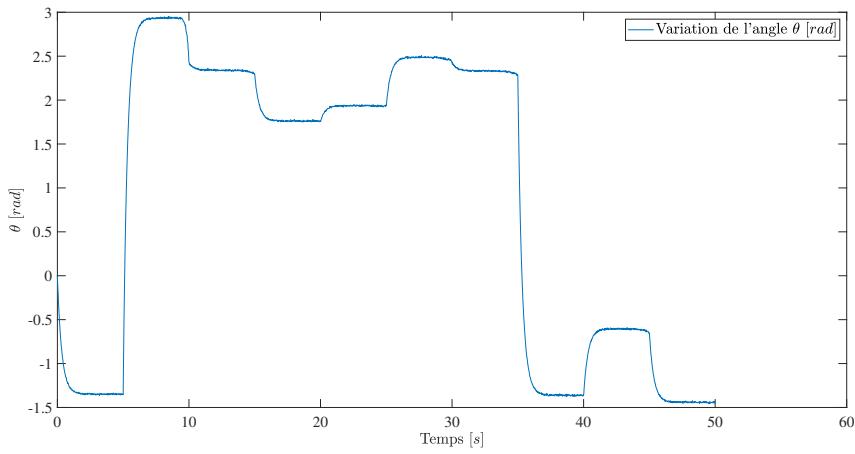


FIGURE II.19 – Chemin de déplacement du robot unicycle $y(x)$ comparé au chemin désiré, en absence de défauts.

Comme on peut le voir sur ces deux figures, à chaque déplacement, le robot atteint parfaitement sa destination désirée, tout en restant très proche du chemin de référence.

Le signal de commande $u = [v, \omega]$ est donné dans la Figure II.20. Quant à la variation de l'angle d'orientation du robot θ , elle est illustrée dans la Figure II.21.

FIGURE II.20 – Signal de commande u .FIGURE II.21 – Variation de l'angle d'orientation du robot θ .

Dans la Figure II.22, un *défaut* (dérive) avec une pente de -10^{-3} rad/s est simulé à l'instant $t = 7$ s au niveau de l'actionneur de la roue droite W_r .

Sur cette Figure II.22, le chemin du robot sans la compensation du *défaut* est représenté en rouge. Sous l'effet du *défaut*, ce dernier s'éloigne de plus en plus du chemin de référence présenté en bleu. Si le *défaut* n'est pas accommodé, le robot risque de devenir incontrôlable et présenter un danger pour les opérateurs et les équipements de la fab. Le chemin du robot après l'accommodation du *défaut* (*i.e.* en utilisant la nouvelle loi de commande tolérante aux fautes, donnée par (II.32)) est présenté sur la même Figure II.22. Comme nous pouvons le remarquer, la nouvelle loi de commande proposée compense parfaitement le *défaut*, et le chemin du robot est similaire à celui présenté dans le mode nominal de la Figure II.18 malgré la présence du *défaut*.

Enfin, afin de montrer que cette nouvelle loi tolérante aux fautes est efficace non seulement pour les *défauts* à faible amplitude, un *défaut* abrupt sous forme d'un échelon avec une amplitude de -1 rad/s a été simulé à l'instant $t = 7$ s au niveau de l'actionneur de la roue gauche W_l . Comme nous pouvons le constater sur la Figure II.23, le *défaut* est parfaitement compensé (courbe donnée en vert), et le chemin du robot après compensation est le même que celui sans *défaut* de la Figure II.18, malgré

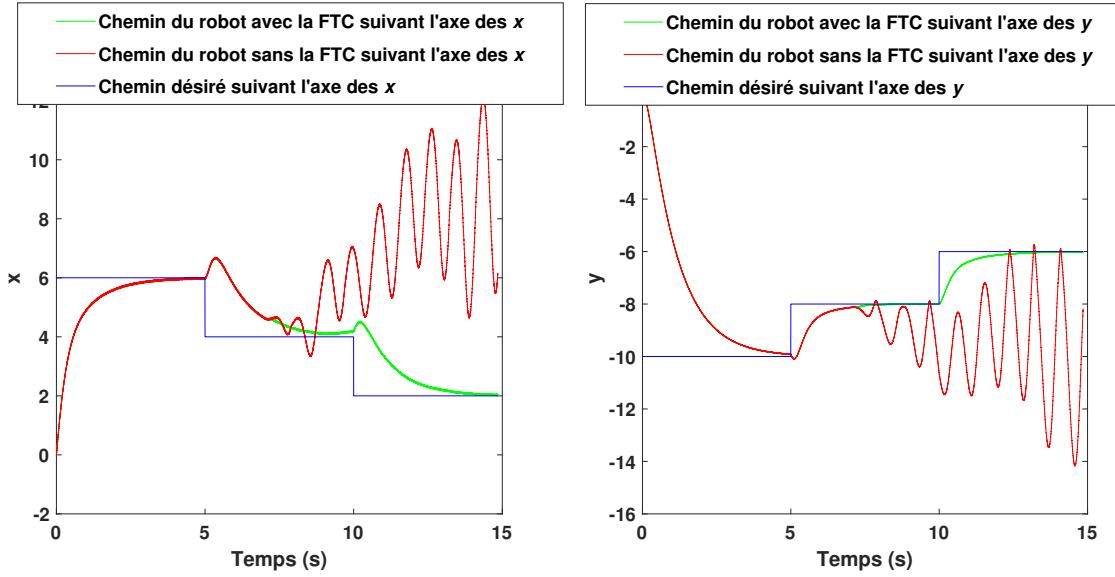


FIGURE II.22 – Chemin du robot en présence d'un *défaut* au niveau de l'actionneur de la roue droite W_r , sans et avec compensation du *défaut*, comparé au chemin désiré.

la sévérité du *défaut* et son impact important sur le comportement du robot (voir la courbe donnée en rouge, sans la compensation du *défaut*).

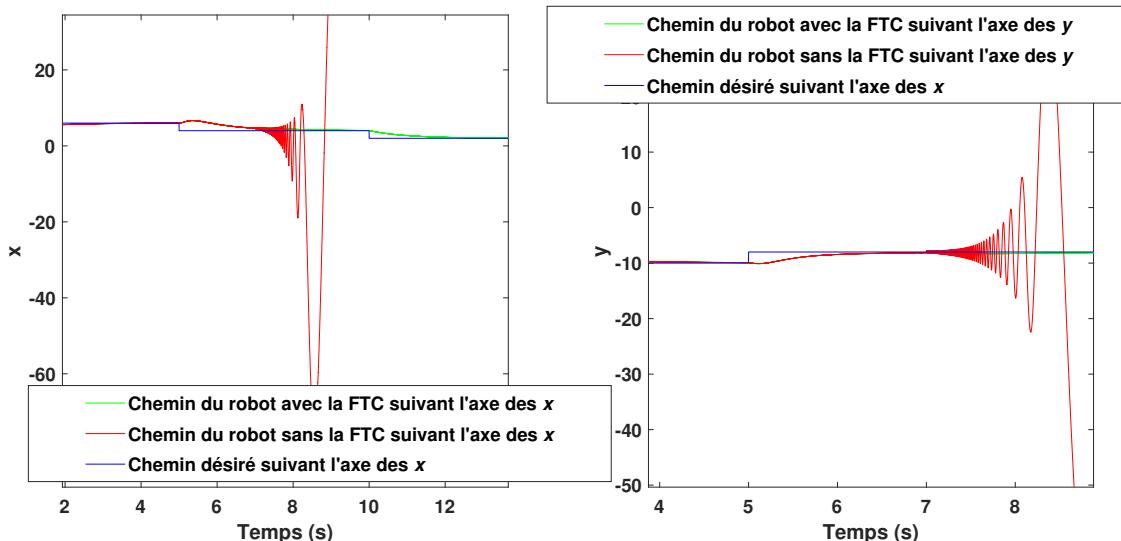


FIGURE II.23 – Chemin du robot en présence d'un *défaut* abrupt au niveau de l'actionneur de la roue gauche W_l , sans et avec compensation du défaut, comparé au chemin désiré.

II.6 Conclusion

Dans ce **Chapitre II**, nous avons présenté les travaux proposés afin de diagnostiquer (*i.e.* détecter, isoler, et estimer) et de compenser les *défauts* des capteurs et des actionneurs que l'on peut avoir sur les robots unicycles intégrés dans la fab.

L'étape du diagnostic repose sur la combinaison de deux approches de diagnostic :

1. La redondance matérielle, en utilisant 3 capteurs au lieu d'un seul pour la mesure de l'orientation du robot dans son espace, noté θ .
2. La redondance analytique, en utilisant des équations analytiques et un banc d'observateurs étendus de Kalman.

La combinaison de ces deux approches nous a permis de générer des résidus ayant une table de signature caractéristique de chaque *défaut*, ce qui facilite l'étape d'isolation. Dans l'étape de l'accommodation et compensation des *défauts*, nous nous sommes particulièrement intéressés aux *défauts* des capteurs qui mesurent la vitesse angulaire des roues, et aux *défauts* des actionneurs de la roue. Le but est d'isoler les *défauts* capteurs des *défauts* actionneurs sachant que l'information de l'état de santé de ces derniers viennent de la même source d'information : les codeurs optiques. Pour compenser l'effet des *défauts* considérés sur le comportement des robots, une approche active basée sur l'estimation et la compensation des *défauts* a été proposée. Les résultats obtenus sont satisfaisants et les figures des résultats de simulation valident l'approche.

Le vrai challenge de ces travaux par rapport à la littérature a été d'un côté l'étude des *défauts* capteurs et actionneurs au même temps, et d'un autre côté la faible amplitude des *défauts* étudiés, qui reflètent le vieillissement des composants qui est une chose inévitable quelles que soient la qualité et la durée de vie des composants.

CHAPITRE III

DIAGNOSTIC DES *défauts* DES ROBOTS OMNIDIRECTIONNELS À 4 ROUES MECANUM

III.1 Introduction	71
III.2 Modélisation des robots omnidirectionnels (4-MWMR)	72
III.2.1 Modèle cinématique	72
III.2.2 Modèle dynamique	74
III.3 Commande des robots omnidirectionnels à 4 roues Mecanum (4-MWMRs)	75
III.4 Diagnostic des <i>défauts</i> capteurs et actionneurs des 4-MWMRs	78
III.4.1 Mesures disponibles et <i>Défauts</i> étudiés	79
III.4.2 Système augmenté	80
III.4.3 Génération des résidus et table de signature	82
III.4.4 Isolation des <i>défauts</i> actionneurs	84
III.4.5 Résultats de simulation	90
III.5 Isolation des <i>défauts</i> actionneurs des 4-MWMRs	92
III.5.1 Principe des observateurs à entrées inconnues (Unknown Input Observers ou UIOs) pour les systèmes linéaires à paramètres variants (LPVS)	93
III.5.2 Génération des résidus et table de signatures	97
III.5.3 Résultats de simulation	99
III.5.4 Sensibilité aux <i>défauts</i> capteurs	102
III.6 Surveillance de l'état de santé des actionneurs des 4-MWMRs et étude de son impact sur le comportement du robot	105
III.6.1 FDI des <i>défauts</i> actionneurs simultanés et identification des <i>défauts</i>	106
III.6.2 Simulation des <i>défauts</i>	107
III.6.3 Détermination des limites de sécurité	109
III.6.4 Résultats de simulation	110
III.6.4.1 Résultats de simulation du FDI des <i>défauts</i> actionneurs simultanés	110
III.6.4.2 Résultats de simulation de la détermination des limites de sécurité	111
III.6.5 Amélioration des limites de sécurité	114
III.6.6 Discussion	117
III.7 Conclusion	118

III.1 Introduction

Grâce à leur nombre de degrés de liberté (DOFs) important, les robots omnidirectionnels sont de plus en plus utilisés dans l'industrie ces dernières années. En effet, comparés aux robots unicycles, les robots omnidirectionnels n'ont pas besoin de faire des manœuvres ou de changer leur direction lors de leur déplacement. Malheureusement, tout comme dans le cas des robots unicycles, l'apparition de *défauts* dans les robots omnidirectionnels est inévitable. La détection et l'isolation de ces *défauts* est donc une étape très importante afin de maîtriser le comportement de ces robots, notamment dans un environnement dynamique tel que la fab où il y a des humains.

Ce chapitre est consacré à la description de trois travaux effectués tout au long de cette thèse [2, 5, 7, 8] afin de répondre aux deux problématiques suivantes :

1. Le diagnostic des *défauts* des capteurs et des actionneurs des robots omnidirectionnels à 4 roues Mecanum utilisés dans la fab.
2. La surveillance du comportement des robots omnidirectionnels à 4 roues Mecanum en fonction de l'évolution de leur état de santé, notamment en présence des *défauts* actionneurs.

Ce chapitre commence par la description des deux modèles cinématique et dynamique des robots omnidirectionnels à 4 roues Mecanum dans la Section III.2. Ensuite, nous présentons dans la Section III.3 la loi de commande utilisée afin de valider nos approches. Cette loi de commande permet aux robots de suivre une trajectoire prédéfinie. Elle permet ainsi de simuler le comportement des robots dans différentes situations, notamment en absence et en présence des *défauts*.

Ensuite, l'approche proposée afin de considérer à la fois les *défauts* capteurs et actionneurs [5] est présentée dans la Section III.4. Cette approche est basée sur l'utilisation d'un observateur étendu de Kalman pour générer des résidus avec des signatures différentes en présence de chaque *défaut* considéré afin de faciliter l'étape d'isolation. Avec cette approche, les *défauts* capteurs sont facilement isolés, tandis que l'isolation des *défauts* actionneurs nécessite de prendre en compte plus de caractéristiques des résidus (*e.g.* l'instant de détection, ...). Cette Section III.4 se termine avec quelques résultats de simulation qui valident l'approche.

Ensuite, afin de faciliter l'isolation des *défauts* actionneurs, une autre approche basée sur un banc d'observateur à entrées inconnues UIOs est proposée dans [2]. Cette approche fait l'objet de la Section III.5, où la méthode est détaillée et des résultats de simulation sont donnés. Enfin, dans la Section III.6, nous décrivons l'approche proposée dans [7, 8] afin de surveiller l'état de santé des actionneurs après une première détection de *défaut*. Le but est de définir des limites de dégradation que les *défauts* (vieillissements) peuvent atteindre au niveau des actionneurs sans que cela n'ait un impact considérable sur le comportement du robot. Cela permet d'éviter l'arrêt immédiat des robots omnidirectionnels après une détection d'un *défaut* au niveau des actionneurs.

III.2 Modélisation des robots omnidirectionnels (4-MWMR)

Généralement, les plate-formes utilisant des roues omnidirectionnelles telle que les roues Mecanum ont trois ou quatre roues actionnées indépendemment. Cependant, les plateformes à trois roues motrices peuvent avoir des problèmes de stabilité liés à la zone de contact avec le sol qui est triangulaire, et à la charge utile qu'elles transportent [187]. Par conséquent, les véhicules à quatre roues sont privilégiés lorsque la stabilité est primordiale. Toutefois, la conduite indépendante de quatre roues pour seulement 3 DOFs rend le robot sur-actionné. Comme mentionné précédemment, nous nous intéressons dans ce manuscrit au robot KUKA KMP (voir Figure I.6, Section I.2.2.2) qui est une plate-forme avec 4 roues Mecanum (nommé par la suite 4-MWMR pour 4 Mecanum wheeled mobile robot). La géométrie de ce robot est illustrée par le schéma de la Figure III.1. Les variables utilisées pour décrire la cinématique et la dynamique de ce robot sont définies dans la TABLE III.1. À noter que les variables $x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}, \theta, \omega, \dot{\omega}, \ddot{\omega}_i$, et τ_i dépendent du temps t . La variable t n'est pas représentée dans les équations pour ne pas alourdir les notations.

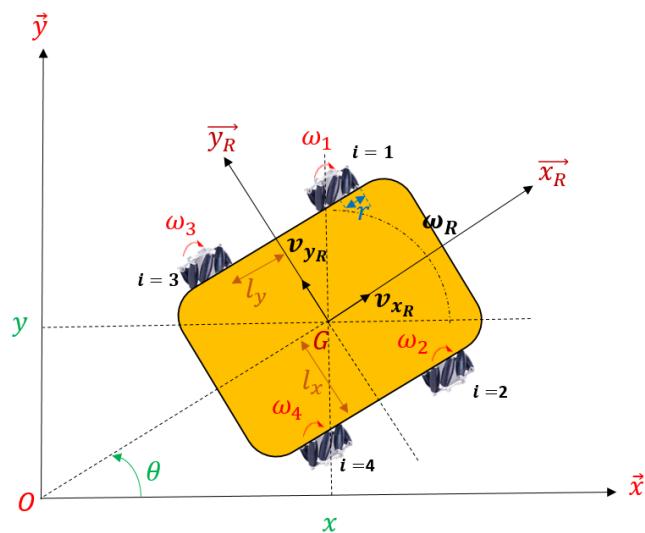


FIGURE III.1 – Géométrie du robot omnidirectionnel.

La modélisation des robots omnidirectionnels, plus particulièrement des plate-formes à 4 roues Mecanum, est un sujet qui a été largement abordé dans la littérature. Parmi les travaux de modélisation de ces robots auxquels on peut se référer, on peut citer [11, 188–195]. Dans cette partie, nous allons décrire le modèle cinématique et dynamique de ces robots. À noter que le fait de considérer le robot tel un point dans son espace, *i.e.* représenter le robot par son centre de masse G , (voir Section II.2) est prise en compte pour la modélisation du robot.

III.2.1 Modèle cinématique

Le modèle cinématique direct (forward kinematic model) des 4-MWMRs vise à exprimer les vitesses du point G (*i.e.* les vitesses du robot) en fonction des vitesses de rotation des 4 roues. Il est

TABLE III.1 – Définition des variables du modèle mathématique du robot omnidirectionnel.

Variable	Description & unité
$(x, y), (x_R, y_R)$	La position du robot suivant \vec{x} et \vec{y} dans les repères $(O, \vec{x}, \vec{y}, \vec{z})$ et $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ respectivement [m].
$(\dot{x} = v_x, \dot{y} = v_y)$	La vitesse en translation du robot suivant \vec{x} et \vec{y} dans le repère $(O, \vec{x}, \vec{y}, \vec{z})$ [m/s].
(\dot{x}_R, \dot{y}_R)	La vitesse en translation du robot suivant \vec{x} et \vec{y} dans le repère $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ [m/s].
$(\ddot{x} = a_x, \ddot{y} = a_y)$	L'accélération en translation du robot suivant \vec{x} et \vec{y} dans le repère $(O, \vec{x}, \vec{y}, \vec{z})$ [m/s ²].
(\ddot{x}_R, \ddot{y}_R)	L'accélération en translation du robot suivant \vec{x} et \vec{y} dans le repère $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ [m/s ²].
θ	L'orientation du robot [rad].
ω	La vitesse angulaire du robot [rad/s].
$\dot{\omega}$	L'accélération angulaire du robot [rad/s ²].
$\dot{\theta}_i$	La vitesse de rotation de la roue i [rad/s].
$\ddot{\theta}_i$	L'accélération angulaire de la roue i [rad/s ²].
l_x	La moitié de la distance entre les roues avant ou les roues arrières (essieu) [m].
l_y	La moitié de la distance entre les roues avant et arrière [m].
$l = l_x + l_y$	
r	Le rayon des roues [m].
I_z	Le moment d'inertie du robot [kg.m ²].
I_w	Le moment d'inertie des roues [kg.m ²].
M	La masse totale du robot [m].
τ_i	Le couple appliqué au moteur de la roue i [N.m].

donné dans le repère local $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ par l'équation (III.1). Le lecteur peut se référer à [188] pour plus de détails (voir le schéma de la Figure III.1 pour la numérotation des roues).

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ \frac{-1}{l} & \frac{1}{l} & \frac{-1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \quad (\text{III.1})$$

En utilisant la matrice de passage (II.1), ce modèle cinématique direct est donné dans le repère global $(O, \vec{x}, \vec{y}, \vec{z})$ sous la forme suivante :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \frac{r}{4} \begin{bmatrix} c+s & c-s & c-s & c+s \\ s-c & s+c & s+c & s-c \\ \frac{-1}{l} & \frac{1}{l} & \frac{-1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = A_r \dot{\theta} \quad (\text{III.2})$$

avec $c = \cos \theta$, $s = \sin \theta$ et $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T$.

III.2.2 Modèle dynamique

Comme pour le cas des robots unicycles, plusieurs approches ont été proposées par les chercheurs afin de modéliser la dynamique des robots omnidirectionnels. Parmi ces approches, on peut citer la méthode du Lagrangien [11, 189–191], la méthode de Newton-Euler [192–195], ou encore l'approche basée sur le Bond Graph [196]. Il est important de préciser que toutes ces approches permettent d'avoir le même modèle, mais décrit de façons différentes.

Le modèle dynamique prend en compte les forces appliquées au robot. Il vise à exprimer le mouvement du robot en fonction des couples appliqués aux moteurs (*i.e.* actionneurs), et il peut être exprimé de différentes façons. Considérer un modèle ou un autre dépend des capteurs dont on dispose, mais aussi des besoins et des objectifs à prendre en compte ensuite pour son utilisation.

Dans cette partie, nous allons présenter deux façons différentes de considérer la dynamique des 4-MWMRs. La première permet de considérer la relation mathématique entre les vitesses angulaires des roues $\dot{\theta}_i$ et les couples moteur τ_i , ($i = 1, \dots, 4$), tandis que la seconde considère la relation mathématique qui lie les accélérations en translation et en rotation ($\ddot{x}, \ddot{y}, \ddot{\theta}$) et les couples moteurs τ_i , ($i = 1, \dots, 4$).

- Le premier modèle considéré dans ce manuscrit est tiré des travaux proposés par [11, 189, 190] en utilisant la méthode du Lagrangien qui tient compte de l'énergie potentielle V et cinétique T du robot dans la modélisation (voir Section II.2.1.3). Dans ce cas d'étude, l'énergie potentielle du robot est $V = 0$ car le robot se déplace sur une surface plane. Quant à son énergie cinétique T , elle est donnée par :

$$T = \frac{1}{2}M(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I_z\omega^2 + \frac{1}{2}I_w(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \quad (\text{III.3})$$

Ensuite, en utilisant la fonction du Lagrange (II.10) (voir Section II.2.1.3) avec $L = T$ comme suit :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) + \frac{\partial L}{\partial \theta} = \tau - \frac{\partial D}{\partial \dot{\theta}} \quad (\text{III.4})$$

on obtient le modèle dynamique du robot comme suit :

$$\tau = M_d\ddot{\theta} + D_\theta\dot{\theta} \quad (\text{III.5})$$

avec $\tau = [\tau_1, \tau_2, \tau_3, \tau_4]^T$, $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T$, $\ddot{\theta} = [\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3, \ddot{\theta}_4]^T$, D_θ étant les coefficients de frottement visqueux des roues qui engendrent une perte d'énergie. Cette perte d'énergie peut être quantifiée en utilisant la relation mathématique suivante : $D = \frac{1}{2}D_\theta(\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2 + \dot{\theta}_4^2)$.

Enfin, la matrice M_d est donnée par :

$$M_d = \begin{bmatrix} a + b + I_w & -b & b & a - b \\ -b & a + b + I_w & a - b & b \\ b & a - b & a + b + I_w & -b \\ a - b & b & -b & a + b + I_w \end{bmatrix} \quad (\text{III.6})$$

$$\text{avec } a = \frac{Mr^2}{8}, b = \frac{I_z r^2}{16l}.$$

- Le deuxième modèle considéré dans les travaux de cette thèse est tiré des travaux proposés par [196], où le Bond Graph est utilisé comme approche de modélisation. Dans ce travail [196], les auteurs divisent la structure globale du robot en sous-domaines tels que le sous-domaine électrique, le sous-domaine mécanique, et le sous-domaine environnemental, et ceci en se basant sur le flux d'énergie à l'intérieur du robot. En utilisant la théorie du Bond Graph, le graphe de chaque sous-système est créé. Ensuite, les graphes de tous les sous-systèmes sont intégrés pour obtenir le modèle dynamique du 4-MWMR. Ce modèle est donné dans le repère $(G, \vec{x}_R, \vec{y}_R, \vec{z}_R)$ comme suit :

$$\begin{cases} M(\ddot{x}_R - \dot{y}_R \omega) + \beta_x \dot{x}_R = \frac{\tau_1 + \tau_2 + \tau_3 + \tau_4}{2r} \\ M(\ddot{y}_R + \dot{x}_R \omega) + \beta_y \dot{y}_R = \frac{\tau_1 - \tau_2 + \tau_3 - \tau_4}{2r} \\ I_z \dot{\omega} = \frac{\tau_1 - \tau_2 - \tau_3 + \tau_4}{2r} l - \beta_\omega \omega \end{cases} \quad (\text{III.7})$$

où $\beta_x, \beta_y, \beta_\omega$ représentent les coefficients du frottement visqueux suivant l'axe longitudinal \vec{x} , l'axe latérale \vec{y} , et l'axe de rotation \vec{z} respectivement.

Suite aux conditions de fonctionnement dans la fab (*i.e.* faible vitesse, faibles perturbations, ...), les frottements visqueux peuvent être négligés. Ainsi, en utilisant la matrice de passage (II.1), le modèle (III.7) est donné dans le repère global $(O, \vec{x}, \vec{y}, \vec{z})$ comme suit :

$$\begin{cases} \ddot{x} = \frac{(c - d)\tau_1 + (c + d)\tau_2 + (c - d)\tau_3 + (c + d)\tau_4}{2Mr} \\ \ddot{y} = \frac{(c + d)\tau_1 + (d - c)\tau_2 + (c + d)\tau_3 + (d - c)\tau_4}{2Mr} \\ \dot{\omega} = \frac{\tau_1 - \tau_2 - \tau_3 + \tau_4}{2I_z r} l \end{cases} \quad (\text{III.8})$$

avec $c = \cos\theta$, et $d = \sin\theta$.

III.3 Commande des robots omnidirectionnels à 4 roues Mecanum (4-MWMRs)

Comme dans le cas des robots unicycles, la commande et le contrôle des robots omnidirectionnels à 4 roues Mecanum (4-MWMRs) ont été largement traités dans la littérature. La commande de ces

robots n'est pas le sujet principal de cette thèse. Néanmoins, nous pouvons citer les trois travaux suivants traitants de ce sujet, ainsi que les références qui s'y trouvent [191, 193, 195].

Dans les travaux de cette thèse, nous proposons d'utiliser la commande par retour d'état linéarisant (Feedback Linearizing Control), proposée par [197]. La théorie de cette commande est basée sur la dérivation par rapport au temps du vecteur de sortie ou de mesures Y_m jusqu'à l'apparition du vecteur de commande u dans l'expression.

Partant de l'équation (III.8), nous obtenons le modèle donné par (III.46) qui sera détaillé dans la Section III.5. Ce modèle peut être commandé en introduisant $\mathcal{Y} = \mathcal{X} = [x, y, \theta]^T$. La dérivée seconde par rapport au temps de $\ddot{\mathcal{X}}$ est une expression linéaire de u avec des paramètres variants. Cette expression est donné par (III.47), Section III.5. En prenant un vecteur v tel que $v = \beta_X u$, on obtient :

$$\ddot{\mathcal{Y}} = \ddot{\mathcal{X}} = \beta_X u = v$$

avec

$$\beta_X = \begin{bmatrix} \frac{c-d}{a} & \frac{c+d}{a} & \frac{c-d}{a} & \frac{c+d}{a} \\ \frac{c+d}{a} & \frac{d-c}{a} & \frac{c+d}{a} & \frac{d-c}{a} \\ \frac{l}{b} & \frac{-l}{b} & \frac{-l}{b} & \frac{l}{b} \end{bmatrix} \quad (\text{III.9})$$

$$c = \cos\theta \text{ et } d = \sin\theta$$

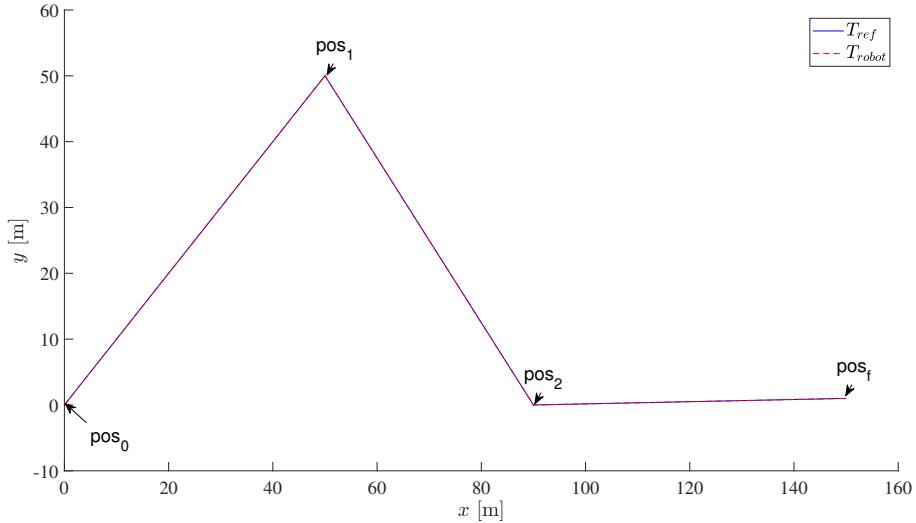
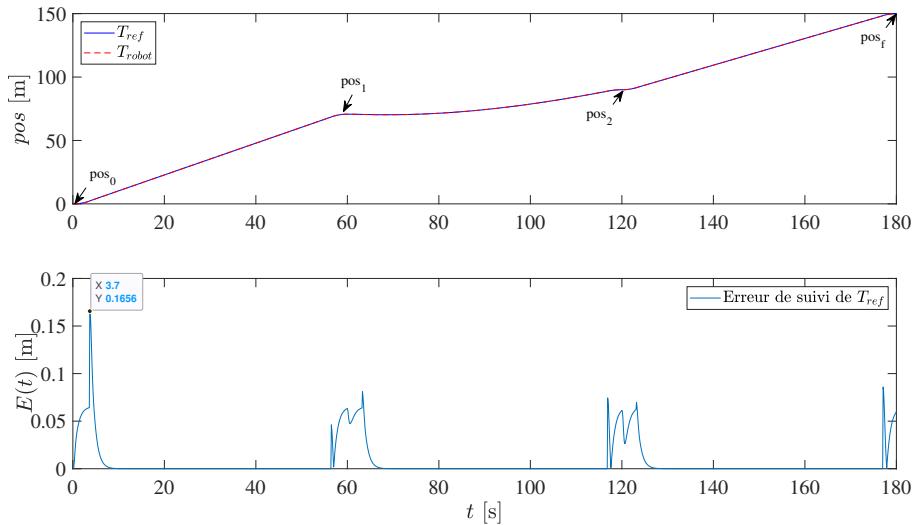
Ensuite, en utilisant la technique de placement de pôles [197], la loi de commande est donnée par :

$$\begin{cases} v &= \sum_{i=0}^{n-1} a_i(Y_{ref}^i - \mathcal{Y}^i) + a_n Y_{ref}^n \\ u &= \beta_X^\dagger v \end{cases} \quad (\text{III.10})$$

où β_X^\dagger dépend de θ et correspond à la pseudo-inverse de Moore-Penrose de β_X , de telle sorte que $\beta_X \beta_X^\dagger = I$, la matrice d'identité de dimension (3×3) , et les coefficients $a_i \in \mathbb{R}$ sont choisis de telle sorte que les pôles du polynôme P défini par $P(s) = \sum_{i=0}^n a_i s^i$ aient des parties réelles négatives, tandis que \mathcal{Y}^i correspond à la $i^{\text{ème}}$ dérivée de ($\mathcal{Y}^0 = \mathcal{Y}$), et Y_{ref}^i à la $i^{\text{ème}}$ dérivée de la trajectoire de référence.

Afin de montrer l'efficacité de cette loi de commande dans le suivi de trajectoire, prenons l'exemple suivant : en partant de la position initiale $pos_0 = [0; 0]^T$, le robot doit atteindre la position finale $pos_f = [150; 1]^T$ en passant par deux positions intermédiaires (*i.e.* en exécutant deux sous-missions) : $pos_1 = [50; 50]^T$ puis $pos_2 = [90; 0]^T$ comme montré sur la Figure III.2. Les coefficients a_i sont pris comme suit : $a_1 = 4$, $a_2 = 3.75$, et $a_3 = 1$.

Comme le montre la Figure III.3, le robot suit parfaitement son chemin prédéfini. L'erreur de suivi $E(t)$ entre la trajectoire de référence Y_{ref} (Reference trajectory) notée T_{ref} , et celle du robot (Robot

FIGURE III.2 – Suivi du profile de trajectoire $y(x)$ en utilisant la commande par retour d'état linéarisant.FIGURE III.3 – Suivi de trajectoire $T_{ref}(t)$ en utilisant la théorie de commande par retour d'état linéarisant.

trajectory), notée T_{robot} est calculée à chaque instant t en utilisant la norme L^2 comme suit :

$$E(t) = \sqrt{|T_{robot}(t)^2 - T_{ref}(t)^2|} \quad (\text{III.11})$$

avec

$$\begin{aligned} T_{robot}(t) &= \sqrt{x(t)^2 + y(t)^2} \\ T_{ref}(t) &= \sqrt{x_{ref}(t)^2 + y_{ref}(t)^2} \end{aligned} \quad (\text{III.12})$$

L'erreur maximale E_m de suivi de trajectoire est définie par :

$$E_m = \max_{t \in t_{mission}} E(t) \quad (\text{III.13})$$

avec $t_{mission}$ la durée planifiée pour la mission (*i.e.* aller du point initial au point final désiré). La valeur maximale obtenue sur E_m est de $E_m = 0,166\text{ m}$. À noter que les performances du suivi dépendent des valeurs prises pour les coefficients a_i .

La Figure III.4 illustre la variation de $u = \tau_i$, ($i = 1, \dots, 4$) calculée en utilisant la loi de contrôle (III.10) afin de suivre le chemin désiré. Quant à la variation de l'angle θ , elle est illustrée dans la

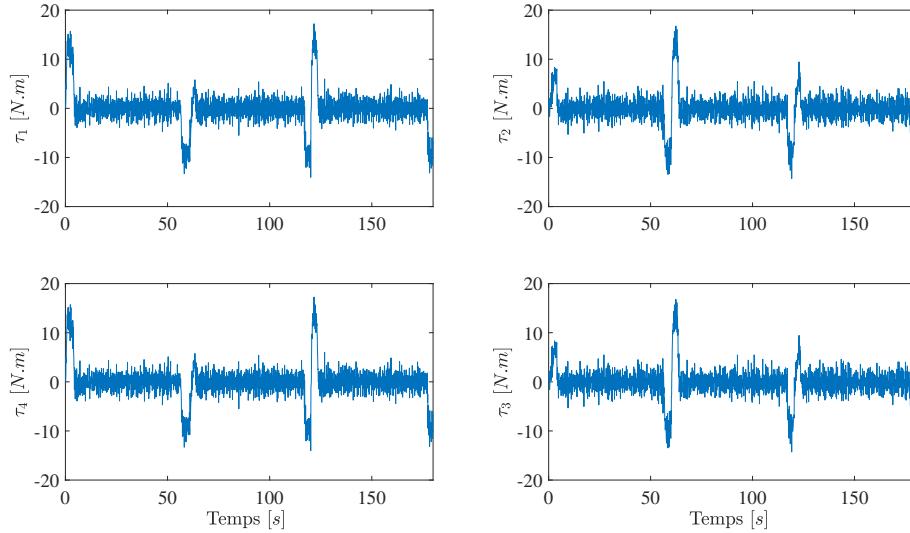


FIGURE III.4 – Évolution du contrôle appliqué au robot.

Figure III.5. Comme nous pouvons le remarquer sur cette figure, θ est constante par sous-mission.

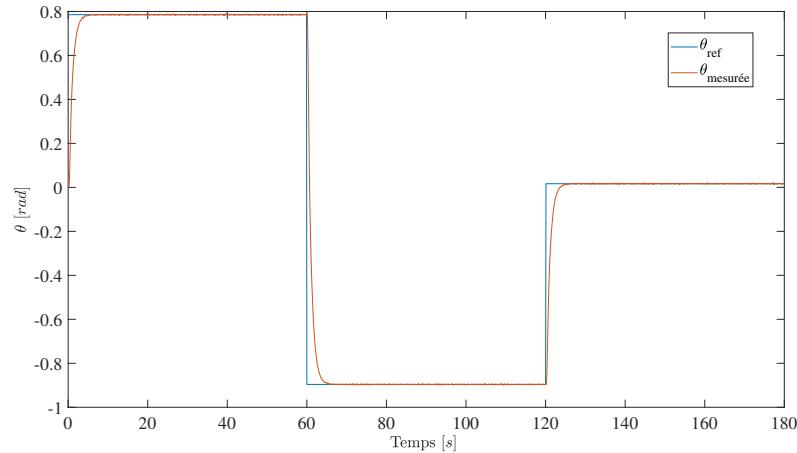


FIGURE III.5 – Variation de l'angle du robot θ .

III.4 Diagnostic des défauts capteurs et actionneurs des 4-MWMRs

Le premier travail sur le diagnostic des *défauts* des robots omnidirectionnels 4-MWMRs proposé dans cette thèse [5] vise à détecter et à isoler à la fois les *défauts* capteurs et actionneurs. Le but est de

signaler tout *défaut* sur ces composants le plus tôt possible afin d'éviter les conséquences indésirables qu'ils peuvent engendrer dans la fab.

Comme illustré précédemment dans la Figure I.7, Section I.2.2.2, les 4-MWMRs sont caractérisés par leur capacité de réaliser différents mouvements dans leur environnement sans avoir besoin de changer leur angle d'orientation θ . La Table III.2 suivante montre la variation des vitesses linéaires ($\dot{x}_R, \dot{y}_R, \omega$) du robot pour quelques mouvements (voir Figure I.7, Section I.2.2.2), ainsi que leurs résultantes sur les vitesses angulaires des roues $\dot{\theta}_i$, ($i = 1, \dots, 4$) selon [188]. c_1 et c_2 étant 2 constantes données pour les vitesses linéaires et angulaires en (m/s) et (rad/s) respectivement.

TABLE III.2 – Le principe de quelques mouvements des robots 4-MWMRs.

		Vitesses						
		\dot{x}_R	\dot{y}_R	ω	$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_3$	$\dot{\theta}_4$
Mouvement	Marche avant	c_1	0	0	c_2	c_2	c_2	c_2
	Marche arrière	$-c_1$	0	0	$-c_2$	$-c_2$	$-c_2$	$-c_2$
	A gauche	0	c_1	0	$-c_2$	c_2	c_2	$-c_2$
	A droite	0	$-c_1$	0	c_2	$-c_2$	$-c_2$	c_2
	Diagonale gauche, marche avant	c_1	c_1	0	0	$2c_2$	$2c_2$	0
	Diagonale gauche, marche arrière	$-c_1$	c_1	0	$-2c_2$	0	0	$-2c_2$
	Diagonale droite, marche avant	c_1	$-c_1$	0	$2c_2$	0	0	$2c_2$
	Diagonale droite, marche arrière	$-c_1$	$-c_1$	0	0	$-2c_2$	$-2c_2$	0

III.4.1 Mesures disponibles et *Défauts* étudiés

Les mesures disponibles sur le robot considéré dans ce travail sont :

- Les vitesses angulaires des 4 roues $\dot{\theta}_i$, ($i = 1, \dots, 4$), mesurées par 4 codeurs optiques placés sur chaque roue.
- L'orientation du robot (*i.e.* l'angle θ), mesurée par un gyroscope.
- La position du robot selon les axes x et y , mesurée par un système de positionnement (*e.g.* caméra, GPS, ...) supposé être fiable et précis.

Quant aux *défauts* étudiés, nous nous intéressons particulièrement aux *défauts* suivants :

a) *Défauts* actionneurs :

1. *Défauts* des actionneurs des quatre roues, notés, A_i , ($i = 1, \dots, 4$) pour désigner respectivement : la roue avant gauche ($i = 1$), la roue avant droite ($i = 2$), la roue arrière gauche ($i = 3$), et la roue arrière droite ($i = 4$) (voir la Figure III.1, Section III.2).

b) *Défauts* capteurs :

1. *Défauts* des quatre codeurs optiques placés sur les quatre roues, notés E_i , ($i = 1, \dots, 4$).
2. *Défauts* gyroscope noté G .

Comme dans le cas des robots unicycles, les *défauts* du système de positionnement du robot ne sont pas considérés car ces mesures sont très importantes pour la navigation autonome des robots. Elles sont donc généralement mesurées avec plusieurs capteurs et outils de positionnement redondants (voir Section I.3.1). La position du robot selon x et y est donc supposée être fiable dans ce travail.

De plus, les *défauts* système ne sont pas considérés car le seul paramètre qui peut varier dans le modèle mathématique du robot (voir (III.2), (III.5), (III.8)) est la masse M , selon si le robot est chargé ou pas, et selon la charge portée. Comme dans le cas des robots unicycles, les variations de la masse du robot en fonction de sa charge sont prises en compte dans la fab.

III.4.2 Système augmenté

Les conditions de fonctionnement dans la fab impliquent que les frottements visqueux lorsque le robot se déplace sont très faibles, voir inexistant. Nous supposons alors que le vecteur des coefficients de frottement visqueux D_θ dans le premier modèle dynamique (voir (III.5), Section III.2) est nul. Cela nous donne :

$$\tau = M_d \ddot{\theta} \quad (\text{III.14})$$

avec

$$M_d = \begin{bmatrix} \alpha & -\beta & \beta & \gamma \\ -\beta & \alpha & \gamma & \beta \\ \beta & \gamma & \alpha & -\beta \\ \gamma & \beta & -\beta & \alpha \end{bmatrix} \quad (\text{III.15})$$

$\alpha = a + b + I_w$, $\beta = b$, and $\gamma = a - b$.

Proposition 1 Pour tout $a > 0$, $b > 0$ et $I_w > 0$, la matrice M_d est inversible.

Preuve de la Proposition 1 : Le déterminant de la matrice M_d est donné par :

$$\det(M_d) = (\alpha - \gamma + 2\beta)(\alpha - \gamma - 2\beta)(\alpha - \gamma)^2 \quad (\text{III.16})$$

M_d est inversible si et seulement si :

$$\det(M_d) \neq 0 \Rightarrow \begin{cases} (\alpha - \gamma + 2\beta) \neq 0 \\ (\alpha - \gamma - 2\beta) \neq 0 \\ (\alpha - \gamma) \neq 0 \end{cases} \quad (\text{III.17})$$

En remplaçant α , β , et γ par leur expression, ces trois conditions s'écrivent alors :

$$\begin{cases} 4b + I_w \neq 0 \\ I_w \neq 0 \\ 2a + I_w \neq 0 \end{cases} \quad (\text{III.18})$$

ce qui est vérifié si $a > 0$, $b > 0$, et $I_w > 0$.

Étant donné que M_d est inversible, nous pouvons alors exprimer le vecteur d'accélération des roues en fonction des couples appliqués aux actionneurs (*i.e.* moteurs) des roues comme suit :

$$\ddot{\theta} = S_d \tau \quad (\text{III.19})$$

avec $S_d = M_d^{-1}$.

Afin d'avoir en simulation toutes les mesures que nous avons sur le vrai robot, le modèle cinématique (III.2) est combiné avec le premier modèle dynamique donné par (III.5) dans la Section III.2, ce qui nous permet d'avoir un système augmenté exprimé dans le repère global $(O, \vec{x}, \vec{y}, \vec{z})$ comme suit :

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \omega \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix}}_X = \underbrace{\begin{bmatrix} 0_{(3 \times 3)} & A_r \\ 0_{(4 \times 7)} & \\ \hline & A_X \end{bmatrix}}_{A_X} X + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ s_{d,11} & s_{d,12} & s_{d,13} & s_{d,14} \\ s_{d,21} & s_{d,22} & s_{d,23} & s_{d,24} \\ s_{d,31} & s_{d,32} & s_{d,33} & s_{d,34} \\ s_{d,41} & s_{d,42} & s_{d,43} & s_{d,44} \end{bmatrix}}_D \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ u \end{bmatrix}}_u \quad (\text{III.20})$$

où A_r est donnée dans (III.2) comme suit : $A_r = \frac{r}{4} \begin{bmatrix} c+s & c-s & c-s & c+s \\ s-c & s+c & s+c & s-c \\ -\frac{1}{l} & \frac{1}{l} & -\frac{1}{l} & \frac{1}{l} \end{bmatrix}$.

Ce modèle peut être écrit sous la forme de représentation d'état suivante :

$$\begin{cases} \dot{X} = A_X X + D u \\ Y = C X z \end{cases} \quad (\text{III.21})$$

avec X le vecteur d'état donné par $X = [x, y, \theta, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T$, A_X la matrice d'état donnée par $A_X = \begin{bmatrix} 0_{(3 \times 3)} & A_r \\ 0_{(4 \times 7)} & \end{bmatrix}$, D la matrice de commande donnée par $D = \begin{pmatrix} 0_{(3 \times 4)} \\ S_d_{(4 \times 4)} \end{pmatrix}$, u le vecteur de commande donné par $u = [\tau_1, \tau_2, \tau_3, \tau_4]^T$, Y le vecteur des mesures. En prenant en considération l'instrumentation du système (voir Section III.4.1), Y est supposé être égal au vecteur d'état X . Par conséquent, la matrice d'observation C est une matrice identité ($I_{7 \times 7}$).

Observabilité du système (III.21) : Compte tenu des capteurs disponibles, tous les états du système X sont mesurés ($X = Y$). De ce fait, le système est observable.

En utilisant la méthode d'Euler, la forme discrète du modèle (III.21) en présence des incertitudes de modélisation $w(k)$ et des bruits de mesures $z(k)$ est donnée par la même expression donnée pour

le robot unicycle avec les matrices appropriées comme suit (voir (II.19), Section II.4.3) :

$$\begin{cases} X(k+1) = A_{d,X}(k)X(k) + Du(k) + w(k) \\ Y(k) = CX(k) + z(k) \end{cases} \quad (\text{III.22})$$

avec $A_{d,X} = I + T_e A_X(k)$ la forme discrète de A_X . Enfin, $w(k)$ et $z(k)$ sont supposés être des bruits gaussiens avec des matrices de covariance connues, notées respectivement Q et S . De plus, $w(k)$ est supposée ne pas avoir de corrélation croisée avec $z(k)$ (voir Section II.4.2).

De même, on obtient la même expression donnée par (II.23) en présence des *défauts* actionneurs et capteurs, avec les paramètres et matrices appropriés au modèle (III.21) :

$$\begin{cases} X(k+1) = A_{d,X}(k)X(k) + Du(k) + E_a f_a(k) + w(k) \\ Y(k) = CX(k) + E_s f_s(k) + z(k) \end{cases} \quad (\text{III.23})$$

E_a et E_s sont les matrices d'incidence des *défauts* actionneurs et capteurs respectivement. Elles sont données par $E_a = D$ et $E_s = C$. Quant à f_a et f_s , ils représentent les vecteurs des *défauts* actionneurs et capteurs respectivement. Ils sont donnés par $f_a = [A_1, A_2, A_3, A_4]^T$ et $f_s = [0, 0, G, E_1, E_2, E_3, E_4]^T$ où A_i représente les *défauts* de l'actionneur de la roue i , ($i = 1, \dots, 4$), E_i les *défauts* du codeur optique i , ($i = 1, \dots, 4$) et G le *défaut gyroscope*.

Remarque : Les éléments A_i du vecteur f_a sont supposés être nuls en absence de *défaut* actionneur, tandis qu'en présence d'un *défaut* sur l'actionneur i , A_i sera non nulle. Idem pour le vecteur f_s , un élément du vecteur n'est différent de zéro qu'en présence d'un *défaut* sur le composant concerné.

Influence du temps d'échantillonage T_e sur l'observabilité du système discret (III.22) :

Le système discret (III.22) perd son observabilité s'il existe deux valeurs propres distinctes λ_1 et λ_2 de la matrice $A_{d,X}$ qui possèdent la même partie réelle et dont la différence des parties imaginaires est un multiple de la pulsation d'échantillonnage $\omega_e = \frac{2\pi}{T_e}$ [177]. En calculant les valeurs propres de la matrice $A_{d,X}$, on trouve 6 valeurs réelles identiques qui valent 1, avec une partie imaginaire nulle. Par conséquent, le système (III.22) est observable quelle que soit T_e . À noter que ce cas d'étude, nous avons pris $T_e = 0.1$.

III.4.3 Génération des résidus et table de signature

La détection et l'isolation des *défauts* actionneurs A_i , ($i = 1, \dots, 4$) et capteurs E_i , ($i = 1, \dots, 4$) et G étudiés dans ce travail sont basées sur la génération de cinq résidus (voir (III.24)) en utilisant un observateur étendu de Kalman (EKO) dont le principe est décrit précédemment dans la Section II.4.3, comme le montre la Figure III.6.

$$\begin{cases} r_1 = \theta - \hat{\theta} \\ r_2 = \dot{\theta}_1 - \hat{\dot{\theta}}_1 \\ r_3 = \dot{\theta}_2 - \hat{\dot{\theta}}_2 \\ r_4 = \dot{\theta}_3 - \hat{\dot{\theta}}_3 \\ r_5 = \dot{\theta}_4 - \hat{\dot{\theta}}_4 \end{cases} \quad (\text{III.24})$$

Pour l'observateur, les matrices de covariance suivantes sont prises :

$$Q = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, S = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

Après la convergence de l'observateur vers la sortie du système, la matrice du gain K suivante est obtenue :

$$K = \begin{bmatrix} 0.9161 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9161 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9161 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.99020 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.99020 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.99020 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.9902 \end{bmatrix}$$

Pour la détection, l'équation (I.1) donnée dans la Section I.4.2.3 est utilisée avec la méthode des 3σ . La signature de ces résidus en présence des différents *défauts* étudiés est décrite dans la Table III.3 :

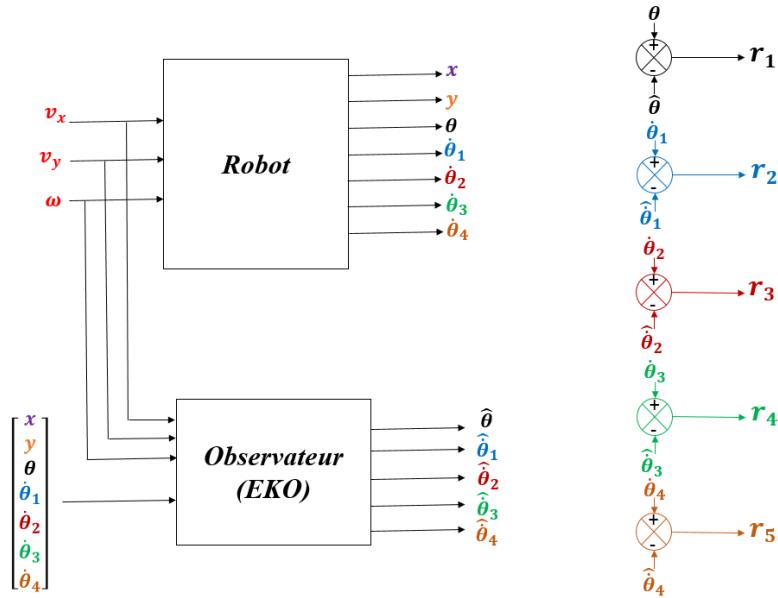


FIGURE III.6 – Génération des résidus pour la détection et l’isolation des *défauts* actionneurs et capteurs du 4-MWMR.

TABLE III.3 – Signature des cinq résidus en présence de *défauts* actionneurs et capteurs.

		Défaut								
		E ₁	E ₂	E ₃	E ₄	G _r	A ₁	A ₂	A ₃	A ₄
Résidu	R ₁	0	0	0	0	1	1	1	1	1
	R ₂	1	0	0	0	0	1	1	1	1
	R ₃	0	1	0	0	0	1	1	1	1
	R ₄	0	0	1	0	0	1	1	1	1
	R ₅	0	0	0	1	0	1	1	1	1

Comme cela peut être constaté dans la Table III.3, les *défauts* capteurs sont détectés et parfaitement isolés. Ils ont des signatures caractéristiques de chaque *défaut*. Quant aux *défauts* actionneurs, ils sont détectés mais pas isolés en se basant sur la signature des résidus uniquement. Les résidus ont la même signature en présence des *défauts* actionneur, quel que soit ce dernier.

Afin d’isoler les *défauts* actionneurs, il est nécessaire de prendre en compte plus de caractéristiques concernant les résidus.

III.4.4 Isolation des *défauts* actionneurs

Dans cette section, en partant de la génération des résidus de la Section III.4.3, nous allons mettre en place une procédure permettant d’isoler les *défauts* actionneurs. Pour cela, nous allons étudier l’impact des *défauts* sur les résidus.

Comme montré dans la démonstration de la Proposition 1, la matrice M_d est inversible. Nous avons :

$$M_d^{-1} = S_d = \begin{bmatrix} \eta & \zeta & -\zeta & \mu \\ \zeta & \eta & \mu & -\zeta \\ -\zeta & \mu & \eta & \zeta \\ \mu & -\zeta & \zeta & \eta \end{bmatrix} \quad (\text{III.25})$$

avec

$$\begin{aligned} \eta &= \frac{I_w^2 + 2ab + aI_w + 3bI_w}{I_w(I_w + 4b)(I_w + 2a)} \\ \zeta &= \frac{b}{I_w(I_w + 4b)} \\ \mu &= -\frac{2ab + aI_w - bI_w}{I_w(I_w + 4b)(I_w + 2a)} \end{aligned}$$

De plus, rappelons que $a = \frac{Mr^2}{8}$ et $b = \frac{I_z r^2}{16l}$. Enfin, seule la masse du robot (chargé/non-chargé) peut changer.

Proposition 2 Lorsque $\frac{20I_w I_z}{16I_w l - r^2 I_z} < M < \frac{80I_w}{r^2}$, nous avons $\eta > 1.1|\mu| > 1.5\zeta$

Démonstration de la Proposition 2 : Nous découpons cette preuve en deux étapes :

- Étape 1 : prouvons que $\eta > 1.1|\mu|$.

En partant des expressions de η et μ , nous avons :

$$\begin{aligned} \frac{\eta}{|\mu|} &= \frac{I_w^2 + 2ab + aI_w + 3bI_w}{2ab + I_w(a - b)} \\ &= 1 + \frac{I_w^2 + 4bI_w}{2ab + (a - b)I_w}. \end{aligned} \quad (\text{III.26})$$

D'un autre côté, nous avons :

$$2ab + (a - b)I_w < 2ab + aI_w \quad (\text{III.27})$$

ce qui nous donne

$$\begin{aligned} \frac{I_w(I_w + 4b)}{2ab + (a - b)I_w} &> \frac{I_w(I_w + 4b)}{a(2b + I_w)} \\ &> \frac{I_w}{a} + \frac{2I_w b}{a(2b + I_w)} \\ &> \frac{I_w}{a} \end{aligned} \quad (\text{III.28})$$

De plus, comme $M < \frac{80I_w}{r^2}$, nous obtenons :

$$I_w > \frac{Mr^2}{80} = \frac{a}{10} \quad (\text{III.29})$$

Finalement, en substituant (III.28) et (III.29) dans (III.26), nous avons $\eta > 1.1|\mu|$.

• **Étape 2 : Prouvons que $|\mu| > 1.5\zeta$.**

Dans ce cas, nous pouvons étudier le quotient $\frac{|\mu|}{\zeta}$. En remplaçant μ et ζ par leurs expressions, nous avons :

$$\begin{aligned} \frac{|\mu|}{\zeta} &= \frac{(2ab + aI_w - bI_w)}{I_w(I_w + 4b)(I_w + 2a)} \frac{I_w(I_w + 4b)}{b} \\ &= \frac{(2ab + aI_w - bI_w)}{b(I_w + 2a)} \\ &= 1 + \frac{I_w(a - 2b)}{b(2a + I_w)} \end{aligned} \quad (\text{III.30})$$

Maintenant si $M > \frac{20I_wI_z}{16I_wl - r^2I_z}$, nous avons

$$4I_wlM > \left(5I_w + 2\frac{Mr^2}{8}\right)I_z \quad (\text{III.31})$$

$$> (5I_wI_z + 2a)I_z \quad (\text{III.32})$$

Cela implique que

$$4I_wlM - 4I_wI_z > I_zI_w + 2aI_z \quad (\text{III.33})$$

En d'autre terme

$$\frac{4I_w}{I_z} \frac{lM - I_z}{I_w + 2a} > 1 \quad (\text{III.34})$$

$$2 \frac{I_w}{b} \frac{a - 2b}{I_w + 2a} > 1 \quad (\text{III.35})$$

Cela veut dire que

$$\frac{I_w}{b} \frac{a - 2b}{I_w + 2a} > \frac{1}{2} \quad (\text{III.36})$$

alors (III.30) devient

$$\frac{|\mu|}{\zeta} > 1.5 \quad (\text{III.37})$$

Nous venons de prouver que $|\mu| > 1.5\zeta$. Cela termine la démonstration de la Proposition 2, et nous avons donc $\eta > 1.1|\mu| > 1.65\zeta$.

□

Lorsque l'actionneur A_i , $i = 1, \dots, 4$ vieillit, cela se traduit par une perte d'efficacité qui peut être modélisée par une dérive sur l'actionneur A_i sous la forme :

$$u_{a,i}(k) = (1 - \gamma_i(k))u_{c,i}(k) \quad (\text{III.38})$$

voir l'équation de perte d'efficacité Section I.4.2, avec $u_{a,i}$ l'entrée exécutée par l'actionneur, et $u_{c,i}$ l'entrée calculée par le contrôleur, et $\gamma_i(k)$ l'amplitude du défaut (le taux de dégradation).

D'une part, en présence de cette dérive, le modèle du système s'écrit :

$$\begin{cases} X(k+1) = A_{d,X}(k)X(k) + Du_a(k) + w(k) \\ Y(k) = CX(k) + z(k) \end{cases} \quad (\text{III.39})$$

autrement dit :

$$\begin{cases} X(k+1) = A_{d,X}(k)X(k) + Du_c(k) + D\Gamma u_c(k) + w(k) \\ Y(k) = CX(k) + z(k) \end{cases} \quad (\text{III.40})$$

avec $D = \begin{pmatrix} 0_{3 \times 4} \\ S_{d,4 \times 4} \end{pmatrix}$ et $\Gamma = \gamma_i(k) \begin{bmatrix} \delta_{1,i} & 0 & 0 & 0 \\ 0 & \delta_{2,i} & 0 & 0 \\ 0 & 0 & \delta_{3,i} & 0 \\ 0 & 0 & 0 & \delta_{4,i} \end{bmatrix}$, $\delta_{j,i} = \begin{cases} 1 & \text{si } j = i \\ 0 & \text{sinon} \end{cases}$

D'une autre part, l'observateur s'écrit sous la forme [82] :

$$\begin{cases} \hat{X}(k+1) = A_{d,X}(k)\hat{X}(k) + Du_c(k) + K(k)(Y(k) - C\hat{X}(k)) \\ \hat{Y}(k) = C\hat{X}(k) \end{cases} \quad (\text{III.41})$$

Ainsi, l'erreur d'estimation $e(k+1)$ devient :

$$\begin{aligned} e(k+1) &= X(k+1) - \hat{X}(k+1) \\ &\approx (A_{d,X}(k) - K(k)C)e(k) + Du_c(k) + w(k) - K(k)z(k) \end{aligned} \quad (\text{III.42})$$

Lemme 1 *Un défaut sur l'actionneur de la roue i est détecté principalement par le résidu r_{i+1} puis par le résidu r_{6-i} .*

Démonstration du Lemme 1 : Les résidus r_2, r_3, r_4 et r_5 sont liés à l'erreur e avec

$$r_i(k) = e_{i+2}(k), i = (2, \dots, 5)$$

De plus, étant donnée l'écriture de l'erreur d'estimation (III.42), et que le gain de l'observateur $K(k)$ est choisi de telle sorte que les pôles de $A_X(k) - K(k)C$ soient à partie réelle strictement négative, nous avons

$$r_i(k) \approx -S_d \Gamma u_c(k) + b(k), i = (2, \dots, 5)$$

avec $b(k)$ le bruit associé à la partie $w(k) - K(k)z(k)$.

Ainsi, lorsque nous avons un *défaut* sur l'actionneur A_i , nous obtenons :

$$r_{i+1}(k) \approx -\gamma_i(k) S_{d,i}^T u_c(k) + b(k)$$

avec $S_{d,i}$ la $i^{\text{ème}}$ colonne de S_d , $i = (1, \dots, 4)$.

Nous pouvons donc voir que la $i^{\text{ème}}$ colonne de S_d a un impact sur les résidus lorsque le $i^{\text{ème}}$ actionneur a un *défaut*.

D'après la Proposition 2 ($\eta > 1.1 |\mu| > 1.65 \zeta$) et l'écriture de S_d , un *défaut* sur l'actionneur A_i aura un impact majeur (poids η) sur le résidu r_{i+1} , un impact moindre sur (poids μ) sur r_{6-i} et enfin un impact mineur (poids ζ) sur les deux derniers résidus. Cela termine la démonstration du Lemme 1.

□

Afin d'isoler les *défauts* en considérant l'impact du *défaut* sur le résidu, nous allons étudier le pourcentage de cet impact en introduisant :

$$r_{i,\%} = 100 \frac{\max |r_i(k)|}{\sum_{i=2}^5 |r_i(k)|}, i = (2, \dots, 5) \quad (\text{III.43})$$

Application sur le robot KUKA KMP :

Les paramètres physiques du robot KUKA-KMP sont donnés dans la Table III.4 suivante :

TABLE III.4 – Les paramètres physiques de la plateforme KUKA-KMR.

Paramètre	Valeur	Unité
l_x	0.1825	m
l_y	0.28	m
r	0.125	m
I_z	50	kg.m ²
I_w	0.89	kg.m ²

Quant à la masse du robot, elle vaut $M = 390 \text{ kg}$ lorsqu'il est non-chargé. Si l'on considère une charge supplémentaire de 100 kg pour le cas où le robot est chargé, la masse vaudra 490 kg .

Tenant compte de ces paramètres, on peut calculer les deux grandeurs suivantes :

$\frac{20I_w I_z}{16I_w l - r^2 I_z} = 153.32$, et $\frac{80I_w}{r^2} = 4557$. Nous pouvons voir que la masse du robot M est comprise

entre ces deux grandeurs que le robot soit chargé ou non. Donc, la Proposition 2 et le Lemme 1 sont vérifiés dans les deux cas. Dans ce qui suit, nous allons apporter une preuve avec des données numériques pour cette approche d'isolation des *défauts* actionneurs.

- **Robot non-chargé** : En remplaçant les paramètres du robot dans (III.25) par leurs valeurs avec $M = 390 \text{ kg}$ (robot non-chargé), la matrice S_d est donnée par :

$$S_d = \begin{bmatrix} 0.6786 & 0.0904 & -0.0904 & -0.2642 \\ 0.0904 & 0.6786 & -0.2642 & -0.0904 \\ -0.0904 & -0.2642 & 0.6786 & 0.0904 \\ -0.2642 & -0.0904 & 0.0904 & 0.6786 \end{bmatrix} \quad (\text{III.44})$$

En utilisant l'équation (III.43), les pourcentages de détection des *défauts* actionneurs par les 4 résidus r_i , ($i = 2, \dots, 5$) sont donnés dans la Table III.5 suivante :

TABLE III.5 – Pourcentage de détection des *défauts* actionneurs par chaque résidus r_i , ($i = 2, \dots, 5$).

Pourcentage de détection/ <i>Défaut</i>	A_1	A_2	A_3	A_4
$r_{2,\%}$	60.28	2.8	2.8	23.32
$r_{3,\%}$	2.8	60.28	23.32	2.8
$r_{4,\%}$	2.8	23.32	60.28	2.8
$r_{5,\%}$	23.32	2.8	2.8	60.28

- **Robot chargé** : Quand le robot est chargé ($M = 490 \text{ kg}$), la matrice S_d devient :

$$S_d = \begin{bmatrix} 0.6497 & 0.0904 & -0.0904 & -0.2931 \\ 0.0904 & 0.6497 & -0.2931 & -0.0904 \\ -0.0904 & -0.2931 & 0.6497 & 0.0904 \\ -0.2931 & -0.0904 & 0.0904 & 0.6497 \end{bmatrix} \quad (\text{III.45})$$

Dans ce cas, les pourcentages de détection des *défauts* sont donnés dans la Table III.6 suivante :

TABLE III.6 – Pourcentage de détection des *défauts* actionneurs par chaque résidus r_i , ($i = 2, \dots, 5$).

Pourcentage de détection/ <i>Défaut</i>	A_1	A_2	A_3	A_4
$r_{2,\%}$	59.62	2.8	2.8	23.98
$r_{3,\%}$	2.8	59.62	23.98	2.8
$r_{4,\%}$	2.8	23.98	59.62	2.8
$r_{5,\%}$	23.98	2.8	2.8	59.62

Nous pouvons bien remarquer que la variation de la masse M n'a pas un impact considérable sur la matrice S_d (on remarque de très petites variations des coefficients de la matrice). Par conséquent, les résidus ainsi que les pourcentages de détection ne sont pas considérablement influencés et l'approche basée sur l'étude de l'impact des *défauts* sur les résidus reste valable quelle que soit la masse du robot

(non-chargé / chargé). Dans ce qui suit, nous allons présenter les résultats de simulation pour le cas du robot non-chargé.

III.4.5 Résultats de simulation

Pour la simulation, les hypothèses suivantes sont faites :

- (H_1) Tous les composants du robot (*i.e.* actionneurs et capteurs) sont en bon état de fonctionnement au début des simulations.
- (H_2) Les *défauts* sont permanents.

Le défaut dans ce travail est d'être capable de détecter et d'isoler parfaitement les *défauts* graduels, qui représentent en pratique le vieillissement des composants. Par conséquent, des *défauts* avec des amplitudes très faibles sont considérés. Ils sont simulés en utilisant des signaux sous forme de rampe avec de faibles amplitudes.

Dans ce qui suit, nous présentons les résultats de simulation du FDI de quelques *défauts* actionneurs et capteurs afin de montrer l'efficacité de l'approche proposée.

Sur la Figure III.7, tous les résidus sont présentés pour illustrer la signature donnée dans la Table III.3 en présence d'un *défaut* au niveau du codeur optique de la seconde roue E_2 . Dans la suite des figures, seuls les résidus significatifs sont présentés. Les résidus restant proches de zéro ne sont pas présentés.

Un signal sous forme d'une rampe variante dans le temps avec une pente de 10^{-2} m/s^2 est utilisé pour simuler le *défaut* sur E_2 . Le signal représente un *défaut* additif avec un pourcentage de 0.01% par rapport à la valeur réelle $\dot{\theta}_2$. Il est détecté et isolé 0.3 s après son apparition comme illustré sur la Figure III.7.

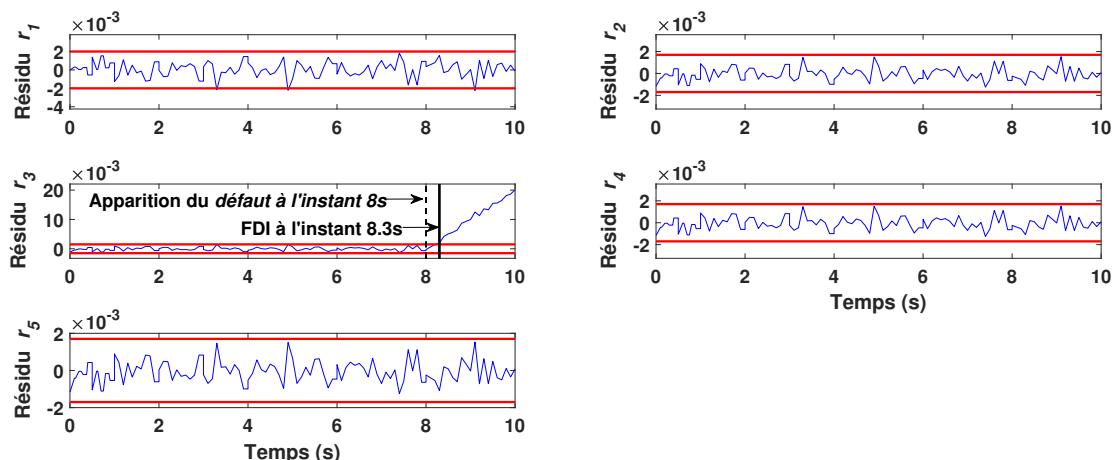


FIGURE III.7 – Signature des 5 résidus en présence d'un *défaut* au niveau du codeur E_2 .

Un *défaut* avec une amplitude de 10^{-2} rad/s est simulé au niveau du gyroscope G . Le résultat du FDI est donné dans la Figure III.8. Comme on peut le voir sur cette figure, le FDI est fait 0.22 s après l'apparition du *défaut*.

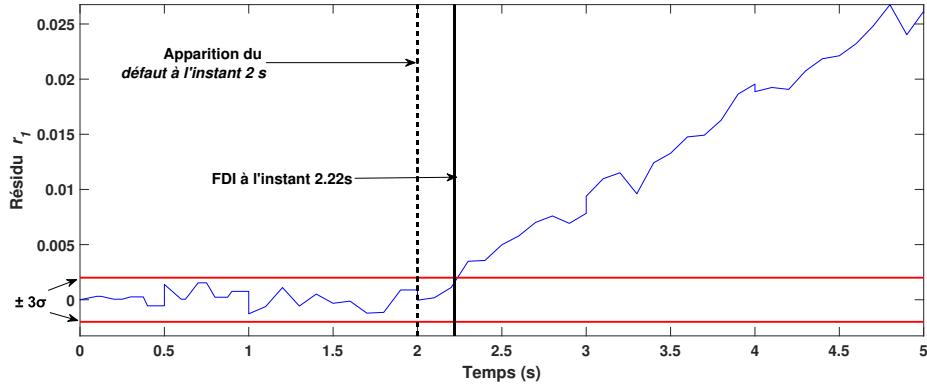


FIGURE III.8 – Résidu r_1 en présence d'un *défaut* gyroscope G .

La Figure III.9 montre le résultat de simulation d'un *défaut* avec une amplitude de 10^{-2} rad/s au niveau de l'actionneur A_1 . Le *défaut* est détecté 6 s après son apparition par le résidu r_2 , puis complètement isolé 0.4 s plus tard.

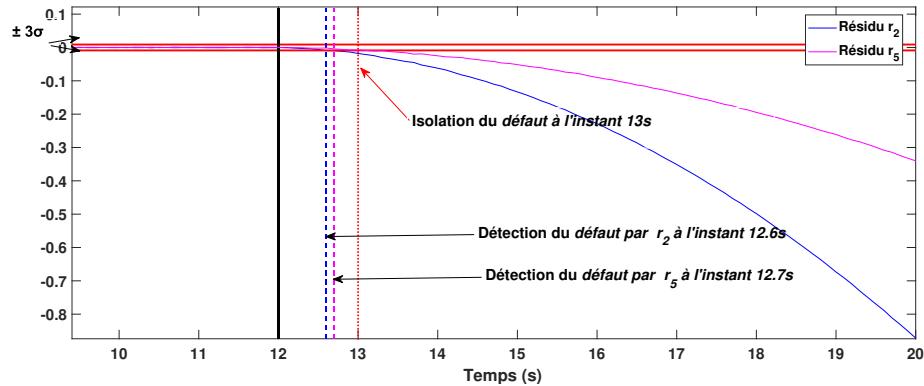


FIGURE III.9 – Résidus r_2 et r_5 en présence d'un *défaut* au niveau de l'actionneur A_1 .

Enfin, un *défaut* avec une amplitude de 10^{-3} rad/s est simulé au niveau de l'actionneur de la troisième roue A_3 . Ce dernier est détecté 1.1 s après son apparition par le résidu r_4 , puis isolé 0.9 s plus tard, comme montré sur la Figure III.10

L'approche basée sur l'observateur étendu de Kalman (EKO) proposée ici permet de détecter et d'isoler très facilement et très rapidement les *défauts* capteurs. Quant aux *défauts* actionneurs, leur isolation n'est pas basée uniquement sur la signature des résidus contrairement aux *défauts* capteurs. Afin de faciliter l'isolation des *défauts* capteurs, une autre approche est proposée dans la Section III.5 suivante.

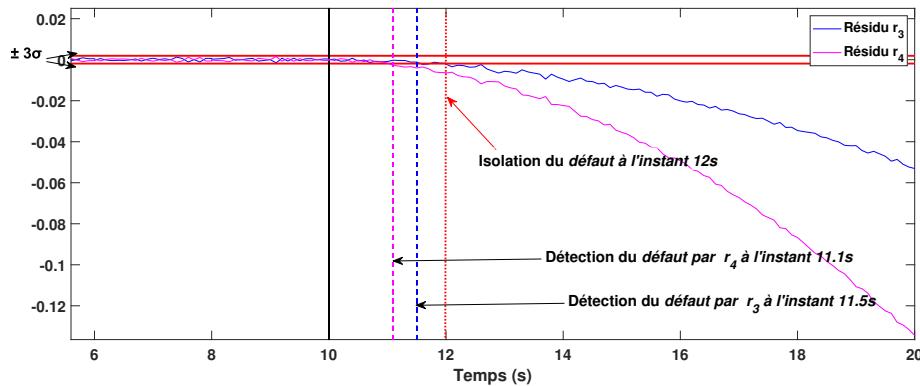


FIGURE III.10 – Résidus r_3 et r_4 en présence d'un défaut au niveau de l'actionneur A_3 .

III.5 Isolation des défauts actionneurs des 4-MWMRs

Comme nous l'avons vu dans la Section III.4 précédente, les *défauts* capteurs considérés sont facilement isolables grâce à la Table de signature des résidus (voir la Table III.3) qui est caractéristique de chaque *défaut* capteur, tandis qu'avec cette approche (*i.e.* Observateur étendu de Kalman EKO), les *défauts* actionneurs sont détectables mais non isolables en se basant uniquement sur la table de signature des résidus. En effet, d'autres caractéristiques portant sur les résidus (*i.e.* la rapidité et le pourcentage de détection) doivent être prises en compte pour l'isolation.

Afin de faciliter l'isolation des *défauts* actionneurs, une autre approche est proposée [2]. La contribution principale de ce travail [2] par rapport au travail précédent [5] mais aussi par rapport à la littérature est :

- L'isolation facile et simple des *défauts* actionneurs en se basant uniquement sur la table de signature des résidus.
- La très faible amplitude des *défauts* considérés dont la précoce détection est un défi.
- La prise en compte de la commande en boucle fermée (III.10). En effet, lorsqu'un *défaut* avec une faible amplitude apparaît au niveau d'un actionneur, il est quasi totalement compensée par la loi de commande en sollicitant davantage les autres actionneurs. De ce fait, l'effet du *défaut* est difficilement identifiable sur le comportement du robot. Mais si ce *défaut* persiste, le reste des actionneurs non défaillants sont fortement sollicités, ils risquent alors de saturer et de vieillir plus vite. Cela peut conduire à l'instabilité ou à l'incontrôlabilité du robot. Cela représente un réel danger pour les opérateurs humains et les équipements de la fab.

Pour des raisons de sécurité et afin éviter la perte du contrôle du robot, tout *défaut* doit être détecté et isolé dans les meilleurs délais, avant que la boucle fermée du robot ne se détériore en un comportement dangereux.

Dans le cas d'un *défaut* actionneur, l'entrée exécutée est différente de l'entrée calculée par le correcteur (l'entrée du robot). Dans ce travail, la théorie des observateurs à entrées inconnues (Unknown Input Observer ou UIO) est utilisée pour le diagnostic des *défauts* actionneurs.

Ce travail est basée sur le deuxième modèle dynamique des 4-MWMRs (III.8) présenté dans la Section III.2. Compte tenu des caractéristiques de ce modèle, la théorie des observateurs à entrées inconnues pour les systèmes à paramètres variants dans le temps (Linear Parameter Varying Systems LPVS) [198] est la plus adaptée. Le but est d'estimer les entrées exécutées par l'actionneur, qui sont supposées être inconnues, puis de les comparer avec celles calculées par le contrôleur pour générer des résidus qui permettent de détecter et d'isoler les *défaux* actionneurs.

III.5.1 Principe des observateurs à entrées inconnues (Unknown Input Observers ou UIOs) pour les systèmes linéaires à paramètres variants (LPVS)

Définition du système : En partant du modèle (III.8) et en prenant le vecteur d'état $X = [x, y, \theta, \dot{x}, \dot{y}, \omega]^T$, on obtient le système donné par la représentation d'état suivante :

$$\begin{cases} \dot{X} = AX + B_X u \\ Y = CX + z \end{cases} \quad (\text{III.46})$$

avec $u = [\tau_1, \tau_2, \tau_3, \tau_4]^T$, z désigne le vecteur des bruits de mesures, supposés être des bruits blancs Gaussiens, non corrélés, et avec une matrice de variance connue. En posant $a = 2Mr$ et $b = 2I_z r$, les matrices A et B_X sont données comme suit :

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_X = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{c-d}{a} & \frac{c+d}{a} & \frac{c-d}{a} & \frac{c+d}{a} \\ \frac{c+d}{a} & \frac{d-c}{a} & \frac{c+d}{a} & \frac{d-c}{a} \\ \frac{l}{b} & \frac{-l}{b} & \frac{-l}{b} & \frac{l}{b} \end{bmatrix} = \begin{bmatrix} 0_{(3 \times 4)} \\ \beta_X \end{bmatrix} \quad (\text{III.47})$$

Quant à la matrice C , elle dépend des mesures disponibles à la sortie du robot. Tenant compte des capteurs embarqués sur le robot KUKA KMP le vecteur Y est donné par $Y = [x, y, \theta, \omega]^T$. Par

conséquent, la matrice C est donnée par :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.48})$$

À noter que les positions selon x et y sont données par un système de positionnement supposé être fiable, l'orientation du robot θ est donnée par un gyroscope, et la vitesse angulaire du robot ω est mesurée à l'aide d'un gyromètre.

Observabilité du système (III.46) : Compte tenu de ces mesures disponibles, le système est observable car la condition nécessaire (III.49) suivante est vérifiée :

$$\text{rang} \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} = n = \dim(X) \quad (\text{III.49})$$

avec $n = 6$. À noter que pour le système (III.46), nous n'avons besoin d'aller jusqu'à CA^{n-1} car en prenant que les deux premiers éléments de cette matrice d'observabilité (III.49), à savoir $\begin{pmatrix} C \\ CA \end{pmatrix}$, la condition d'observabilité est vérifiée, nous obtenons :

$$\text{rang} \begin{pmatrix} C \\ CA \end{pmatrix} = \text{rang} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = 6$$

Réécriture du système et aspects théoriques des observateurs à entrées inconnues pour les systèmes LPV : Partant du modèle dynamique (III.46), $B_X u$ peut être écrit comme $\mathcal{B}_X \mathbf{u} + D_X d$. Par conséquent, (III.46) devient :

$$\begin{cases} \dot{X} = AX + \mathcal{B}_X \mathbf{u} + D_X d \\ Y = CX + z \end{cases} \quad (\text{III.50})$$

où \mathbf{u} est un vecteur d'entrée de dimension (3×1) , supposé être connu et calculé par la loi de commande par placement de pôles (III.10) donnée dans la Section III.3, \mathcal{B}_X la matrice de contrôle qui lui correspond, d une des 4 entrées (scalaire) supposée être inconnue, et D_X le vecteur de contrôle lui correspondant et qui correspond à une colonne de B_X .

En se basant sur la théorie des UIOs pour les systèmes LPV [198], l'observateur associé au système donné par (III.50) est donné par :

$$\begin{cases} \dot{Z} = N_X Z + G_X \mathbf{u} + H_X Y \\ \hat{X} = Z - E_X Y \\ \hat{Y} = C \hat{X} \end{cases} \quad (\text{III.51})$$

À noter que les matrices $N_{X(6 \times 6)}$, $G_{X(6 \times 3)}$, $H_{X(6 \times 4)}$ et $E_{X(6 \times 4)}$ ne sont pas constantes et leur variation dépend de l'angle d'orientation θ qui est une partie du vecteur d'état X . Définissons l'erreur d'estimation d'état comme suit :

$$\begin{aligned} e &= X - \hat{X} \\ &= X - Z + E_X Y \\ &= (I_n + E_X C)X - Z + E_X z \\ &= P_X X - Z + E_X z \end{aligned} \quad (\text{III.52})$$

La dérivée temporelle de l'erreur d'estimation est donnée par :

$$\begin{aligned} \dot{e} &= P_X \dot{X} + \dot{P}_X X - \dot{Z} \\ &= N_X e + (P_X A - N_X P_X - H_X C + \dot{P}_X)X + (P_X \mathcal{B}_X - G_X) \mathbf{u} + P_X D_X d - H_X z \end{aligned} \quad (\text{III.53})$$

L'UIO existe pour le système (III.50) si l'erreur tend vers 0 lorsque le temps tend vers l'infini ($e \rightarrow 0$ quand $t \rightarrow \infty$). Ainsi, les conditions suivantes doivent être vérifiées :

- (C1) $N_X e$ est asymptotiquement stable
- (C2) $P_X A - N_X P_X - H_X C + \dot{P}_X = 0$
- (C3) $P_X \mathcal{B}_X - G_X = 0$
- (C4) $P_X D_X = 0$

À noter également que les deux conditions nécessaires suivantes [88] sont satisfaites en tenant compte du nombre de mesures disponibles sur le robot :

- La condition du rang : $\text{rang}(CD_X) = \text{rang}(D_X)$.
- La paire (A_1, C) doit être observable, où $A_1 = (AP_X + \dot{P}_X) - D_X[(CD_X)^T CD_X]^{-1}(CD_X)^T CA$

Enfin, l'entrée inconnue d peut être estimée en utilisant l'équation (III.54) suivante [88] :

$$\hat{d} = (CD_X)^\dagger [\dot{Y} - CA\hat{X} - C\mathcal{B}_X \mathbf{u}] \quad (\text{III.54})$$

Mise en oeuvre de l'observateur à entrées inconnues (UIO) : Les étapes de résolution de l'observateur se résument comme suit :

1. Partant de la condition (C4), nous avons $(I_n + E_X C)D_X = 0$. Donc $E_X = -D_X(CD_X)^\dagger$ où $(CD_X)^\dagger = (CD_X)^T(CD_X(CD_X)^T)^{-1}$ est l'inverse généralisé de (CD_X) . $(CD_X)^\dagger$ existe si et seulement si la condition du rang est satisfaite.
2. Connaissant E_X , déduire $P_X = I_n + E_X C$.
3. Partant de (C3), calculer $G_X = P_X \mathcal{B}_X$.
4. Fixer les valeurs propres de N_X de telle sorte qu'elles aient une partie réelle strictement négative. N_X est donnée par la diagonale des poles.
5. Partant de (C2), et posons $K_X = H_X + N_X E_X$, nous obtenons $N_X = P_X A - K_X C + \dot{P}_X$.
6. Si la paire $(P_X A + \dot{P}_X, C)$ est observable, calculer K_X pour que les valeurs propres de $(P_X A + \dot{P}_X - K_X C)$ soient égales à celles de N_X définies en 4.
7. Calculer $H_X = K_X - N_X E_X$.

Application sur un cas : Dans le cadre du 4-MWMR et dans le cas où l'on considère que l'entrée exécutée par le premier actionneur A_1 est inconnue (différente de celle calculée τ_1), les matrices \mathcal{B}_X

et D_X sont données par : $\mathcal{B}_X = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{c+d}{a} & \frac{c-d}{a} & \frac{c+d}{a} \\ \frac{d-c}{a} & \frac{c+d}{a} & \frac{d-c}{a} \\ \frac{-l}{b} & \frac{-l}{b} & \frac{l}{b} \end{bmatrix}$, $D_X = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{c-d}{a} \\ \frac{a}{c+d} \\ \frac{a}{l} \end{bmatrix}$. Dans ce cas, l'observateur est

construit comme suit :

$$1. E_X = \begin{bmatrix} 0_{(3 \times 4)} & -b(c-d) \\ 0 & 0 & 0 & \frac{al}{-b(c+d)} \\ 0 & 0 & 0 & \frac{al}{al} \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$2. P_X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{-b(c-d)}{al} \\ 0 & 0 & 0 & 0 & 1 & \frac{-b(c+d)}{al} \end{bmatrix}, \quad \dot{P}_X = \begin{bmatrix} 0_{(3 \times 6)} & b\dot{\theta}(c+d) \\ 0 & 0 & 0 & 0 & 0 & \frac{al}{b\dot{\theta}(d-c)} \\ 0 & 0 & 0 & 0 & 0 & \frac{al}{al} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$3. G_X = P_X D_X = \begin{bmatrix} 0_{(2 \times 3)} & 2d \\ \frac{2c}{2d} & \frac{2(c-d)}{2(c+d)} & \frac{a}{-2c} \\ \frac{a}{a} & \frac{a}{a} & \frac{a}{a} \\ 0 & 0 & 0 \end{bmatrix}$$

4. Le vecteur des pôles suivants sont pris : $\text{poles} = [-0.2943, -0.8883, -0.2, -0.8338, -0.904, -0.3252]^T$

5. Calculer le gain K_X qui permet d'avoir les valeurs propres de

$$P_X A + \dot{P}_X - K_X C = \begin{bmatrix} -K_{11} & -K_{12} & -K_{13} & 1 & 0 & -K_{14} \\ -K_{21} & -K_{22} & -K_{23} & 1 & 0 & -K_{24} \\ -K_{31} & -K_{32} & -K_{33} & 0 & 0 & 1 - K_{34} \\ -K_{41} & -K_{42} & -K_{43} & 0 & 0 & \frac{b\dot{\theta}(c+d)}{al} - K_{44} \\ -K_{51} & -K_{52} & -K_{53} & 0 & 0 & \frac{b\dot{\theta}(d-c)}{al} - K_{54} \\ -K_{61} & -K_{62} & -K_{63} & 0 & 0 & -K_{64} \end{bmatrix}$$

égales au vecteur poles .

6. Déduire N_X .

7. Calculer H_X .

Rendre $N_X e$ asymptotiquement stable équivaut à choisir la matrice N_X avec des valeurs propres à partie réelle non positive. Plus ces dernières sont petites, plus la convergence de l'observateur est rapide. La Figure III.11 illustre l'estimation de la première entrée \hat{u}_1 , comparée à celle calculée $u_{1,cal}$ en absence de *défaut*. On voit clairement sur cette Figure III.11 que l'observateur converge rapidement.

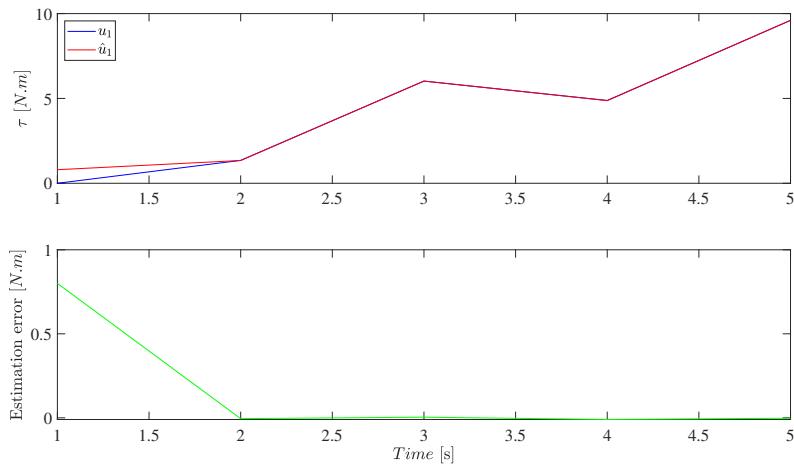


FIGURE III.11 – Illustration de la convergence du UIO vers la valeur réelle et son erreur d'estimation.

III.5.2 Génération des résidus et table de signatures

Afin de pouvoir détecter et isoler tous les *défauts* actionneurs étudiés, un banc d'observateurs à entrées inconnues sous la structure généralisée GOS pour le cas des *défauts* actionneurs (voir Figure I.16, Section I.4.2.3) est utilisé tel qu'illustré sur la Figure III.12.

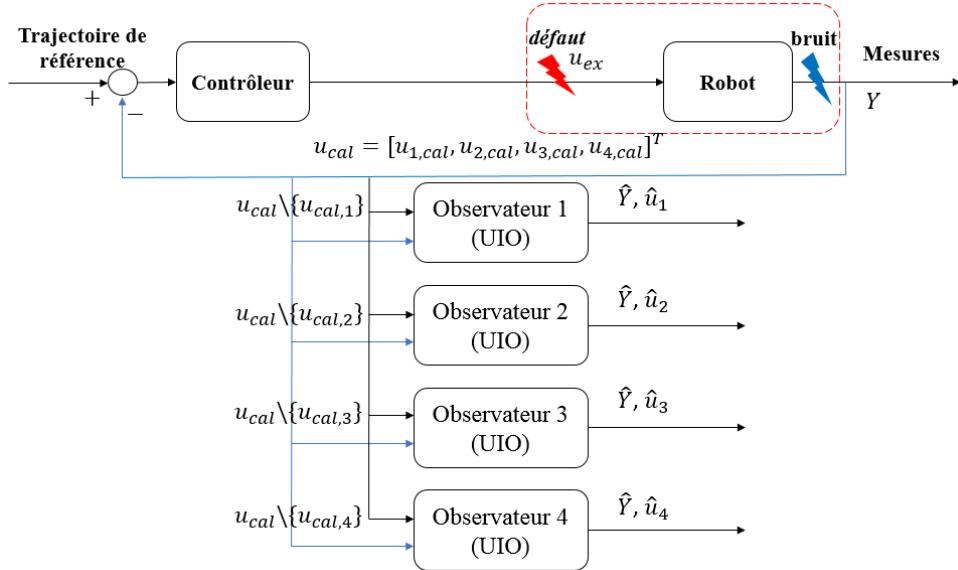


FIGURE III.12 – Banc d’observateurs UIOs sous la structure GOS pour le FDI des défauts actionneurs.

Quand un *défaut* apparaît, il impacte les sorties Y du système. Ainsi, en utilisant les sorties du système Y comme entrée de tous les observateurs, les entrées inconnues, correspondant à u_{ex} sur la Figure III.12, sont estimées avec *défaut*.

Comme montré sur la Figure III.12, le but est d’utiliser l’observateur i , ($i = 1, \dots, 4$) afin d’estimer l’entrée $u_{i,ex}$ qui ne fait pas partie des entrées de ce dernier. Ainsi, en comparant les entrées calculées du système u_{cal} (*i.e.* les sorties du contrôleur) avec les entrées estimées \hat{u}_i , quatre résidus sont générés comme suit :

$$\left\{ \begin{array}{l} r_1 = u_{1,cal} - \hat{u}_1 \\ r_2 = u_{2,cal} - \hat{u}_2 \\ r_3 = u_{3,cal} - \hat{u}_3 \\ r_4 = u_{4,cal} - \hat{u}_4 \end{array} \right. \quad (\text{III.55})$$

Comme dans le cas des travaux précédents, des seuils fixes sont calculés en utilisant la règle des 3σ [89], et l’algorithme (I.1) est utilisé afin de réduire les fausses alarmes. La signature des résidus décisionnels R_i en présence des *défauts* actionneurs notés A_i , ($i = 1, \dots, 4$) est donnée dans la Table III.7 suivante :

TABLE III.7 – Signature des résidus décisionnels en présence des *défauts* actionneurs.

		Défaut				
		\emptyset	A_1	A_2	A_3	A_4
Résidu	R_1	0	1	0	0	0
	R_2	0	0	1	0	0
	R_3	0	0	0	1	0
	R_4	0	0	0	0	1

Comme le montre la Table III.7, le fait d'utiliser une banc d'UIOs permet non seulement de détecter les *défauts* actionneurs, mais aussi de les isoler parfaitement et d'une façon efficace.

III.5.3 Résultats de simulation

Les paramètres physiques du robot sont donnés précédemment dans la Table III.4. Le bruit de mesure est modélisé en prenant en compte la précision de chaque capteur donnée comme suit :

- Les positions x et y sont données par le système de positionnement avec une précision de ± 5 mm.
- θ est donnée par le gyroscope avec une précision de $\pm 0.5^\circ$.
- La vitesse ω est donnée par le gyromètre avec une précision de $\pm 0.3^\circ/s$.

À noter que les deux hypothèses (H_1) et (H_2) faites précédemment dans la Section III.4.5 restent valables dans ce travail (les conditions initiales n'ont pas de *défaut*, et les *défauts* sont permanents après apparition).

Nous nous intéressons particulièrement au vieillissement et à la perte d'efficacité des actionneurs suite à la sollicitation continue des robots dans la fab. Comme mentionné précédemment, ces *défauts* sont généralement caractérisés par des amplitudes assez faibles. Dans ce qui suit, les résultats de détection et d'isolation de quelques *défauts* simulés sur les actionneurs, avec des amplitudes différentes sont présentés.

En absence de *défauts*, si on prend en considération les résidus r_i , on détecte quelques fausses alertes. Par contre, en tenant compte des résidus décisionnels R_i générés en utilisant l'équation de détection (I.1), Section I.4.2.3 tous les résidus décisionnels sont à zéro et les fausses alertes sont donc pas prises en compte (voir Figure III.13). Encore une fois on montre que l'algorithme de détection est robuste à ces fausses alertes.

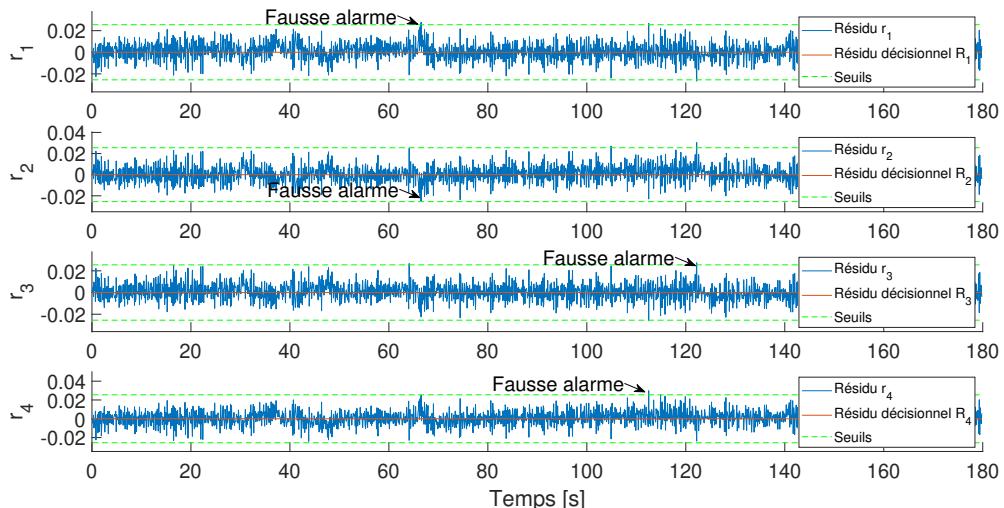


FIGURE III.13 – Les résidus en mode nominal, absence de *défauts*.

Dans ce qui suit, et pour des raisons de clarifications, seuls les résidus significatifs des *défauts* sont présentés. Le reste des résidus restants proche de zéro ne sont pas considérés.

La Figure III.14 montre le FDI d'un *défaut* simulé au niveau de l'actionneur de la roue 1 (*i.e.* A_1). Ce *défaut* représente le vieillissement de l'actionneur dans le temps. L'entrée exécutée par l'actionneur est : $u_{1,ex} = \alpha(t - t_a)$ où l'amplitude α est proportionnelle à $u_{1,cal}$ et est donnée par $\alpha = 2.5e^{-5}u_{1,cal}$, t représente l'instant courant et t_a l'instant d'apparition du *défaut*. Le *défaut* est simulé à l'instant 6 s et il est détecté et isolé à l'instant 13.3 s (voir la Figure III.14).

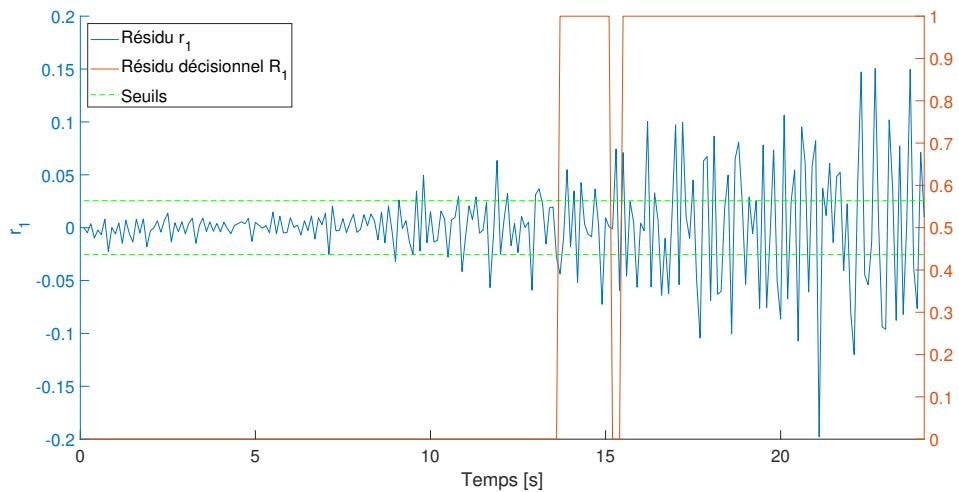


FIGURE III.14 – Le résidu r_1 en présence d'un *défaut* actionneur A_1 .

Dans la Figure III.15, l'entrée exécutée de l'actionneur de la roue 2 est donnée par $u_{2,ex} = (1-\beta)u_{2,cal}$, $\beta = 2e^{-3}$ à partir de l'instant 36 s. Ce *défaut* représente la perte d'efficacité de l'actionneur avec un pourcentage de 0.2% de la valeur de l'entrée calculée. Malgré sa faible amplitude, le *défaut* est détecté et isolé par r_2 0.6 s après son apparition.

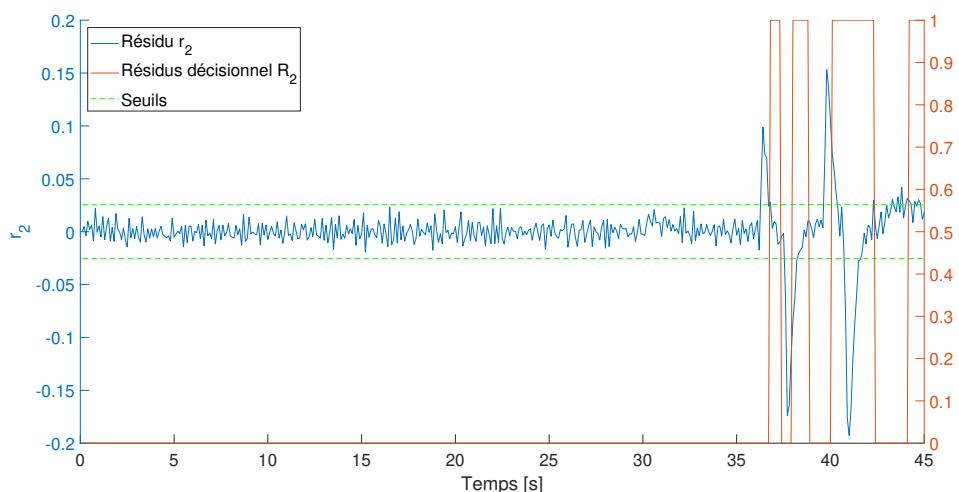


FIGURE III.15 – Le résidu r_2 en présence d'un *défaut* actionneur A_2 .

Un autre *défaut* est simulé au niveau de l'actionneur de la roue 3 à l'instant 30 s. Ce dernier est donné par $u_{3,ex} = (1 - \gamma)u_{3,cal}$, $\gamma = 1e^{-3}$ et il représente la perte de 0.1% de l'efficacité de l'actionneur A_3 . Il est détecté et isolé 6 s après son apparition, comme montré sur la Figure III.16.

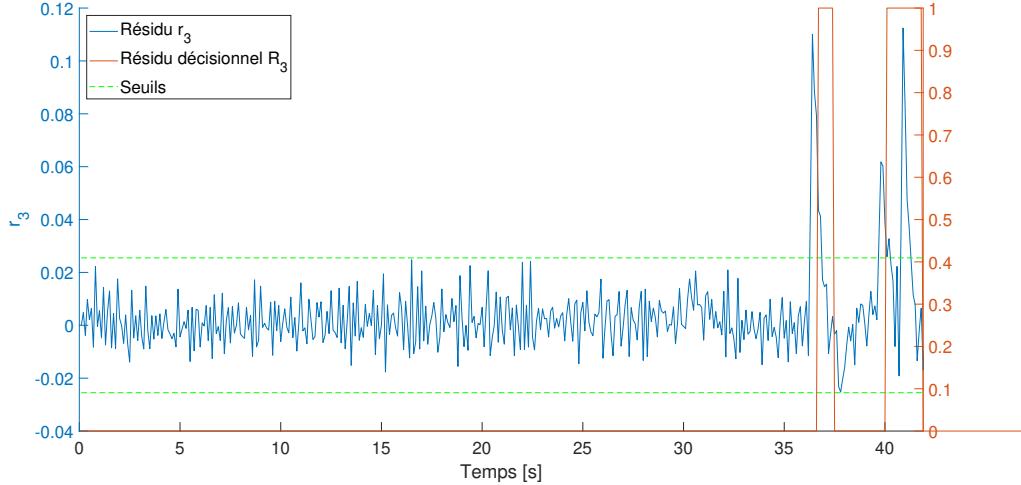


FIGURE III.16 – Le résidu r_3 en présence d'un *défaut* actionneur A_3 .

Enfin, un *défaut* représentant le vieillissement de l'actionneur de la roue 4 est simulé. Il est donné par $u_{4,ex} = u_{4,cal} - \zeta(t - t_b)$, $\zeta = 1e^{-4}u_{4,cal}$, avec t l'instant courant et $t_b = 30$ s l'instant d'apparition du *défaut*. Le FDI de ce *défaut* est fait juste 1 s après son apparition (voir la Figure III.17).

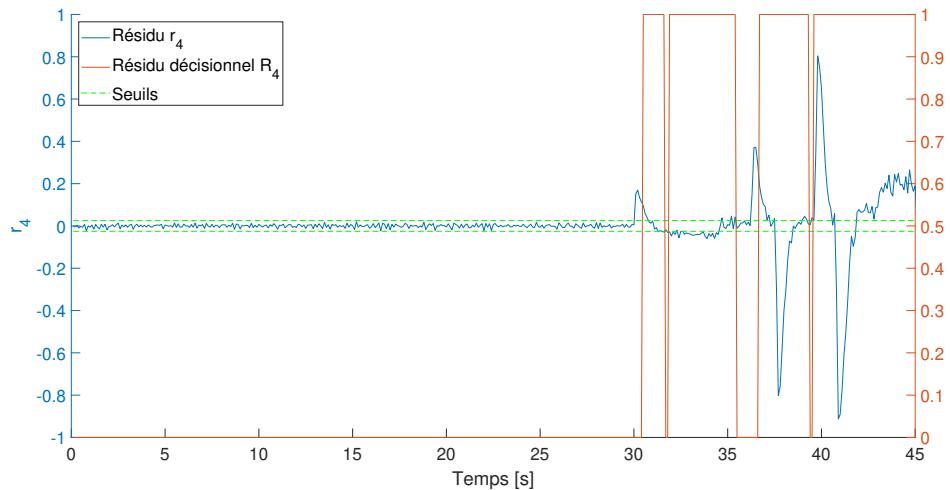


FIGURE III.17 – Le résidu r_4 en présence d'un *défaut* actionneur A_4 .

Comme on peut le remarquer sur les quatre figures (Figure III.14 et Figure III.17), parfois le *défaut* est détecté pendant une petite période du temps puis il n'est plus détecté malgré sa présence permanente (voir (H_2) , Section III.4.5). Quelques secondes plus tard, il est de nouveau détecté. Ceci est lié au fait que les *défauts* simulés sont proportionnels aux entrées calculées $u_{i,cal}$, ($i = 1,..,4$) des actionneurs. Selon le mouvement exécuté par le robot, parfois tous les actionneurs ne sont pas

sollicités en même temps (voir Table III.2, Section III.4). Par conséquent, quand un actionneur n'est pas sollicité, c'est-à-dire que sa commande vaut zéro ($u_{i,cal} = 0$, $i = 1, \dots, 4$), son *défaut* n'est pas détecté.

III.5.4 Sensibilité aux *défauts* capteurs

Afin d'étudier l'impact des *défauts* capteurs sur les quatre résidus générés avec cette approche à base des UIOs (*i.e.* en absence de *défauts* actionneurs, et quand un *défaut* capteur apparaît, les quatre résidus vont-ils être impactés?), nous allons calculer la formule d'estimation de l'entrée inconnue \hat{d} en présence des *défauts* actionneurs et capteurs.

D'un côté, le système (III.50) s'écrit en présence des *défauts* capteurs et actionneurs (d correspond à l'entrée exécutée par l'actionneur, et qui peut contenir le *défaut*) comme suit :

$$\begin{cases} \dot{X} &= AX + \mathcal{B}_X \mathbf{u} + D_X d \\ Y &= CX + E_s d_s + z \end{cases} \quad (\text{III.56})$$

où $E_s = I_{(4 \times 4)}$ est la matrice d'incidence des *défauts* capteurs, et d_s le vecteur des *défauts* capteurs donné par $d_s = [d_x, d_y, d_\theta, d_{\dot{\theta}}]^T$ où d_x, d_y représentent respectivement un *défaut* sur la position x et y , d_θ un *défaut* gyroscope et $d_{\dot{\theta}}$ un *défaut* gyromètre.

D'un autre côté, nous avons d'après l'équation (III.54)

$$\hat{d} = (CD_X)^\dagger [\dot{Y} - CA\dot{X} - C\mathcal{B}_X \mathbf{u}]$$

Pour calculer $\dot{Y} = C\dot{X} = C(\dot{X} - \dot{e})$, nous calculons tout d'abord \dot{e} en présence des *défauts* capteurs comme suit :

$$\dot{e} = N_X e + (P_X A - N_X P_X - H_X C + \dot{P}_X) X + (P_X \mathcal{B}_X - G_X) \mathbf{u} + P_X D_X d + (N_X E_X - H_X) E_s d_s - H_X z$$

En satisfaisant les conditions (C1), ..., (C4), \dot{e} devient :

$$\dot{e} = N_X e + (N_X E_X - H_X) E_s d_s - H_X z \quad (\text{III.57})$$

En substituant (III.57) dans $\dot{Y} = C(\dot{X} - \dot{e})$ on obtient :

$$\dot{Y} = C(AX + \mathcal{B}_X u + D_X d - N_X e - (N_X E_X - H_X) E_s d_s + H_X z) \quad (\text{III.58})$$

En remplaçant \dot{Y} par (III.58) dans (III.54), on obtient enfin :

$$\hat{d} = (CD_X)^\dagger C(A - N_X)e + d + (CD_X)^\dagger C(H_X - N_X E_X) F_s d_s \quad (\text{III.59})$$

Comme nous pouvons le voir dans cette expression (III.59), les *défauts* capteurs d_s impactent l'estimation de d via le facteur $(CD_X)^\dagger C(H_X - N_X E_X)F_s$. En calculant ce facteur, on obtient $[0, 0, 0, s]$, avec s une constante qui dépend des paramètres physiques du robot (*i.e.* M, r, I_z). En conclusion, seul le *défaut* du gyromètre peut impacter les estimations $\hat{d}_i = \hat{u}_i$ des observateurs UIOs, et donc les résidus (III.55). De plus, pour que ce *défaut* ait un impact sur les estimations, il faut qu'il soit d'une amplitude assez élevée. Jusqu'à une amplitude de l'ordre $0.05\text{rad/s}^2 = 2.86^\circ/\text{s}^2$, le *défaut* n'a aucun impact sur les estimations. Un *défaut* avec cette amplitude est un *défaut* assez élevé qu'il faut qu'il soit détecté bien avant (avec l'approche proposée dans le travail précédent [5]).

Afin d'illustrer cela, la Figure III.18 représente les quatre résidus en présence des *défauts* capteurs suivants :

1. Un *défaut* graduel au niveau de la position selon x : $x_{\text{measured}} = x_{\text{real}} + 0.9(t - t_s)$ durant l'intervalle $[3 \ 8] \text{s}$, t_s étant l'instant d'apparition du *défaut* (*i.e.* 3s).
2. Un biais au niveau de la position selon y : $y_{\text{measured}} = y_{\text{real}} + 1.5$ durant l'intervalle $[9 \ 14] \text{s}$.
3. Le gyroscope est supposé avoir un *défaut* de calibration : $\theta_{\text{measured}} = 0.1\theta_{\text{real}}$ durant l'intervalle $[15 \ 20] \text{s}$.
4. Le gyromètre est supposé être bloqué sur une mesure : $\omega_{\text{measured}} = \text{constant}$ durant l'intervalle $[21 \ 27] \text{s}$. La constante correspond à la valeur de la vitesse angulaire $\omega = 0.0035\text{rad/s}^2$ à l'instant 21s , sachant que sans le *défaut*, cette mesure varie de $3.5e^{-3}\text{rad/s}^2$ à l'instant $t = 21 \text{s}$ à $6.23e^{-4}$ à l'instant $t = 27 \text{s}$.

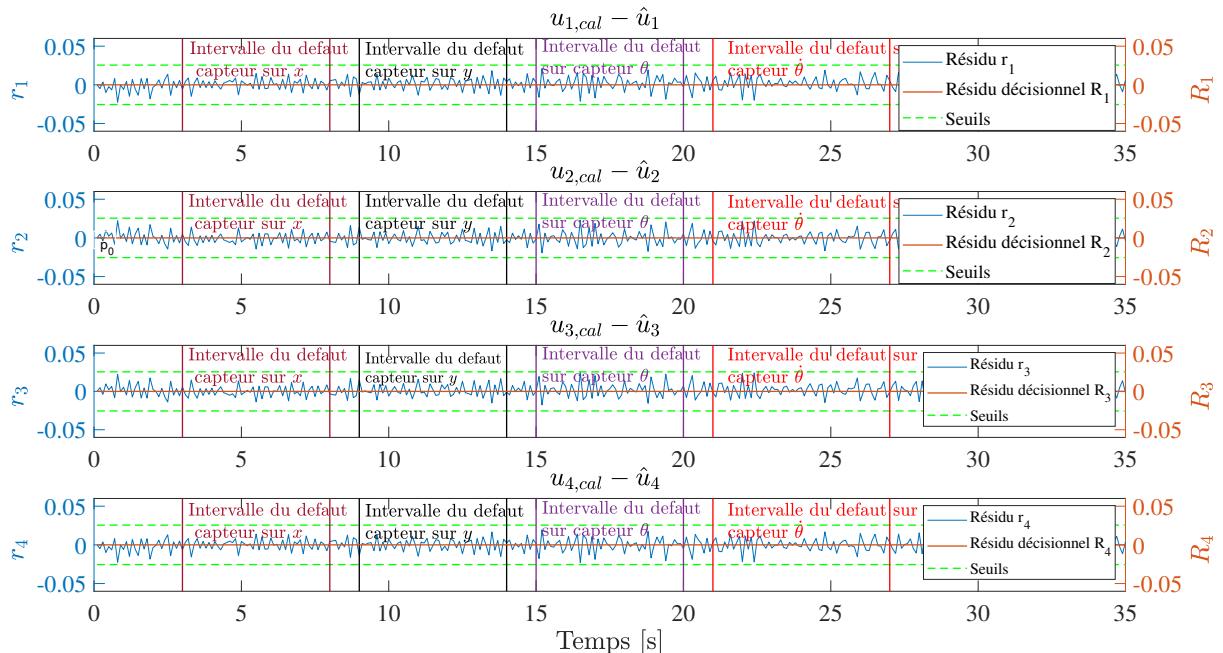


FIGURE III.18 – Les quatre résidus en présence de *défauts* capteurs.

À rappeler que suite à la fiabilité et la redondance des systèmes de positionnement utilisés sur les robots mobiles (voir Section I.3.1), les *défauts* sur les positions x et y ne sont pas considérés dans cette thèse (voir Section III.4.1). Néanmoins, afin de montrer la robustesse de l'approche proposée ici, nous étudions l'impact de ces *défauts* sur les résidus générés. En effet, dans le cas où un *défaut* sur le système de positionnement apparaît (malgré la faible probabilité d'apparition), ce *défaut* n'a aucun impact sur les résidus, tout comme les *défauts* sur le gyroscope, comme montré dans la Figure III.18. À noter également que nous avons simulé des *défauts* qui ont un impact considérable (*défauts* importants) sur le comportement du robot afin de bien montrer que les *défauts* capteurs, quelle que soit leur sévérité, n'impactent pas les résidus générés avec cette approche. Quant aux *défauts* gyromètre, il faut qu'ils soient assez importants (le capteur n'est plus fonctionnel) pour qu'ils impactent les résidus.

Afin de montrer l'effet de ces *défauts* capteurs sur le comportement du robot, la trajectoire du robot en absence et en présence de ces derniers est illustrée sur la Figure III.19, où le robot part du point initial $pos_0 = [0; 0]^T$ au point désiré $pos_f = [50; 50]^T$. La trajectoire de référence (Reference trajectory) notée $T_{ref}(t)$ et celle du robot (Robot trajectory) notée $T_{robot}(t)$ sont respectivement calculées par les équations données par (III.12), Section III.3, tandis que l'erreur du suivi (Tracking Error) notée E est calculée à chaque instant t en utilisant l'équation (III.11), Section III.3. L'erreur maximale E_m obtenue sur le suivi de trajectoire, définie par (III.13), Section III.3 est de l'ordre de $E_m = 0.16 \text{ m}$ en absence des *défauts* capteurs, tandis qu'elle atteint environ $E_m = 5 \text{ m}$ en leur présence. On montre bien que ces *défauts* ont un impact considérable sur le comportement du robot.

Comme nous pouvons le remarquer sur la Figure III.19, malgré l'importance des *défauts* capteurs simulés et leur impact sur le comportement du robot, les quatre résidus restent proches de zéro (voir la Figure III.18). L'approche est robuste aux *défauts* capteurs.

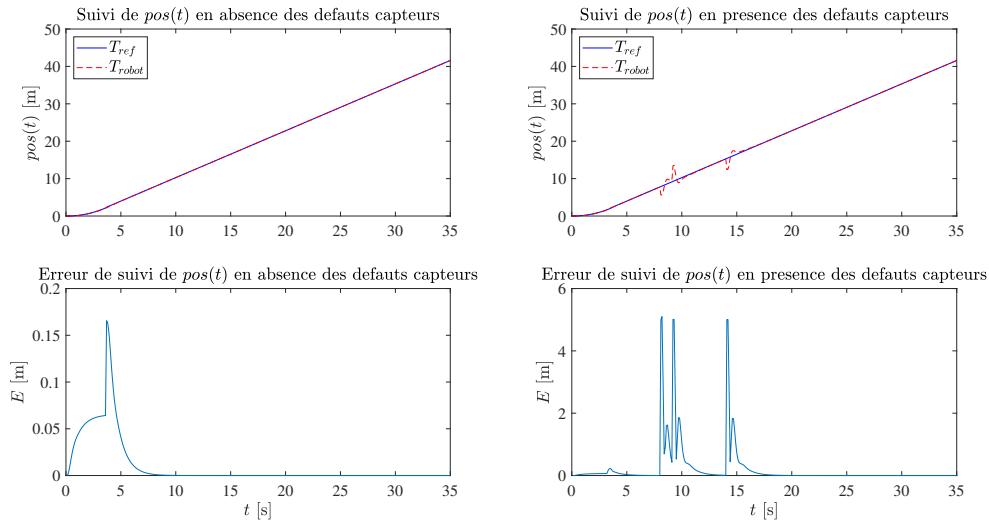


FIGURE III.19 – Suivi de trajectoire en présence des *défauts* capteurs.

III.6 Surveillance de l'état de santé des actionneurs des 4-MWMRs et étude de son impact sur le comportement du robot

La détection et l'isolation des *défauts* actionneurs est une étape très importante surtout quand il s'agit des robots destinés à évoluer dans des milieux collaboratifs, ce qui est le cas dans le cadre de cette thèse. Dans certaines conditions de fonctionnement, mais aussi en fonction de la charge et de la mission du robot, le risque d'apparition d'un *défaut* actionneur augmente considérablement. Ces derniers peuvent avoir un impact significatif sur le comportement du robot et présenter un réel danger pour le robot lui-même ainsi que pour les personnes et les équipements à proximité [199].

Le diagnostic précoce de ces *défauts* permet de prévoir des solutions avant la défaillance totale de l'actionneur et avant que le *défaut* ait un impact considérable sur le comportement du robot. Généralement, et dans les applications industrielles, deux solutions principales sont prises en compte après une première détection d'un *défaut* dans un système :

S_1 Commutation de la boucle fermée vers une commande tolérante aux fautes (FTC) si c'est possible.

S_2 Arrêt immédiat du système pour une maintenance corrective.

Concernant la commande tolérante aux fautes (FTC) pour les 4-MWMRs, il a été prouvé dans plusieurs références bibliographiques dont on peut citer [11, 199], que grâce à la redondance cinématique du robot (*i.e.* quatre actionneurs pour seulement trois DOFs), le robot peut fonctionner avec trois roues sans avoir recours au changement de la loi de commande. Le FTC n'est pas nécessaire avant d'atteindre le cas où deux actionneurs sont complètement défectueux (*i.e.* ils ne sont pas en mesure de recevoir des commandes du contrôleur, mais les roues peuvent toujours tourner librement autour de son arbre).

Néanmoins, dans le cas de perte du contrôle de deux actionneurs à la fois, des approches FTC actives et passives (voir Figure II.10) ont été proposées (voir [11, 199, 200] et leurs références). Par exemple, selon la configuration des deux actionneurs défectueux, les auteurs de [11] réadaptent le modèle du robot à un modèle d'un robot unicycle et une nouvelle loi de commande est calculée par la suite. Plus récemment, un modèle FTC prédictif est proposé dans [199].

Pour le cas de perte complète de deux actionneurs à la fois, les résultats des approches FTC proposées dans la littérature sont prometteurs. Mais dans l'industrie, notant que la probabilité de laisser un actionneur se détériorer jusqu'à la perte complète de son contrôle est très faible, la probabilité d'atteindre le cas où deux actionneurs sont complètement incontrôlables sans aucune intervention préalable est proche de zéro. Néanmoins, et surtout lorsque le robot est fortement sollicité, des *défauts* simultanés au niveau des actionneurs dus à l'usure ou au vieillissement, sont plus susceptibles d'apparaître avant d'atteindre le niveau de la perte complète du contrôle de ces actionneurs. Ces *défauts* sont souvent décrits sous forme d'une perte d'efficacité de l'actionneur.

Malgré le fait que la probabilité d'avoir une dégradation affectant simultanément plus d'un actionneur soit supérieure à celle de perdre complètement et simultanément le contrôle de deux actionneurs, le cas traitant les dégradations simultanées n'est pas étudié dans la littérature.

Note 1 : Le terme «simultané» est utilisé ici pour désigner les *défaux* qui peuvent apparaître durant la même période mais pas nécessairement et rigoureusement au même instant : un deuxième *défaut* apparaît avant que le premier ne soit traité.

Après une première détection d'un ou de plusieurs *défaux* actionneurs, et connaissant les conséquences de ces derniers, la deuxième solution S_2 mentionnée précédemment est plus susceptible d'être adaptée dans l'industrie. Cependant, arrêter immédiatement le robot pour une maintenance corrective après une détection d'un *défaut* n'est pas toujours possible. Cela peut impacter le rendement de la production d'une manière significative, notamment lorsque l'amplitude des *défaux* n'est pas importante et n'a pas d'impact critique sur le comportement du robot.

En effet, afin d'éviter l'arrêt en urgence du robot lorsque cela n'est pas nécessaire, nous proposons dans ce travail une méthodologie permettant de superviser l'évolution de l'état de santé des actionneurs après la détection de *défaut*, et d'étudier l'impact de cet état de santé sur le comportement du robot. Les *défaux* considérés sont la perte d'efficacité des actionneurs. Deux cas principaux sont étudiés :

- **Cas 1 :** Les 4 actionneurs perdent de 0 % (*i.e.*, aucun *défaut*) à 95 % de leur efficacité.
- **Cas 2 :** Un actionneur est complètement défectueux, tandis que les 3 autres perdent de 0 % à 95 % de leur efficacité. Le terme "actionneur complètement défectueux" est utilisé pour désigner un actionneur non contrôlable, mais les roues peuvent tourner librement autour de son arbre de moyeu.

Note 2 : Le **Cas 2** est étudié ici car il peut se produire accidentellement malgré sa faible probabilité.

Le but derrière cette étude est de définir des intervalles de sécurité décrivant les limites de dégradation tolérées afin de voir quel niveau de détérioration peut être atteint au niveau des actionneurs tandis que le robot continue de fonctionner en respectant les consignes de sécurité imposées par son environnement de travail. Cela vise à fournir à l'utilisateur un support décisionnel (*i.e.* Decision Making Support) pour savoir si l'arrêt immédiat du robot est nécessaire ou non. Si non, dans quelles conditions autoriser le robot à continuer de fonctionner.

III.6.1 FDI des *défaux* actionneurs simultanés et identification des *défaux*

Il est à noter que l'approche proposée dans le travail précédent (Section III.5) permet également d'isoler les *défaux* simultanés, (voir la Table III.7 qui montre bien que la signature des résidus est caractéristique de chaque *défaut*). Les premières figures de la Section III.6.4.1 illustrent les résultats d'isolation de quelques *défaux* simultanés.

Quant à l'identification des *défaux*, qui se définit par l'estimation du pourcentage de perte d'efficacité de chaque actionneur noté P_i , ($i = 1, \dots, 4$), il est déduit en utilisant la relation mathématique

suivante :

$$P_i[\%] = 100 \left(1 - \frac{\hat{u}_i}{u_{i,cal}} \right) \quad (\text{III.60})$$

où \hat{u}_i représente l'entrée estimée, image de l'entrée exécutée ($u_{i,ex}$) par l'actionneur de la roue i , estimée par l'observateur i (voir Figure III.12), et $u_{i,cal}$ l'entrée calculée par le contrôleur pour l'actionneur i .

Dans la Figure III.20, l'actionneur de la roue 1, (A_1) est supposé perdre 10 % de son efficacité. L'estimation du *défaut* en utilisant l'équation (III.60) est illustrée en comparaison avec l'efficacité restante pour l'actionneur suivant deux cas différents :

- En absence du *défaut*.
- En présence du *défaut*.

On voit très bien via cet exemple que les *défauts* sont parfaitement estimés.

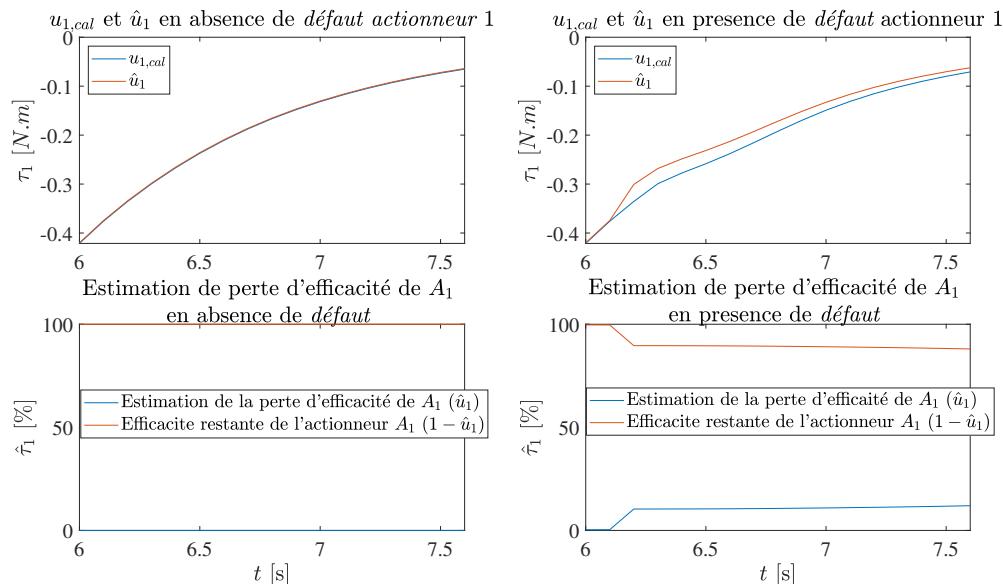


FIGURE III.20 – Estimation de la perte d'efficacité de l'actionneur A_1 .

III.6.2 Simulation des *défauts*

Comme mentionné précédemment, deux cas sont étudiés : **Cas 1** et **Cas 2**. L'actionneur complètement défectueux est simulé en remplaçant la colonne lui correspondant dans la matrice B_θ dans (III.47) par des 0, ou en ajoutant un défaut avec une perte d'efficacité de 100 %.

Pour le premier cas d'étude, des simulations sont effectuées en ajoutant simultanément aux actionneurs (pas nécessairement au même instant) des dégradations allant de 0 % (pas de dégradation) à 95 % avec un pas de 5 %. Ensuite, pour étudier l'impact de ces dégradations sur le comportement du robot, l'écart ou l'erreur de suivi E entre la trajectoire de référence T_{ref} et la trajectoire du robot T_{robot}

est calculée, et seule sa valeur maximale E_m (III.13), est enregistrée à chaque simulation. Ceci est effectué en utilisant l'algorithme de l'organigramme de la Figure III.21 où $u_{i,cal}$ et $u_{i,ex} = \hat{u}_i$ désignent les entrées calculées et exécutées respectivement. Ensuite, l'algorithme donné par l'organigramme de la Figure III.21 est utilisé pour le deuxième cas d'étude (**Cas 2**), et l'erreur maximale E_m est enregistrée à chaque exécution.

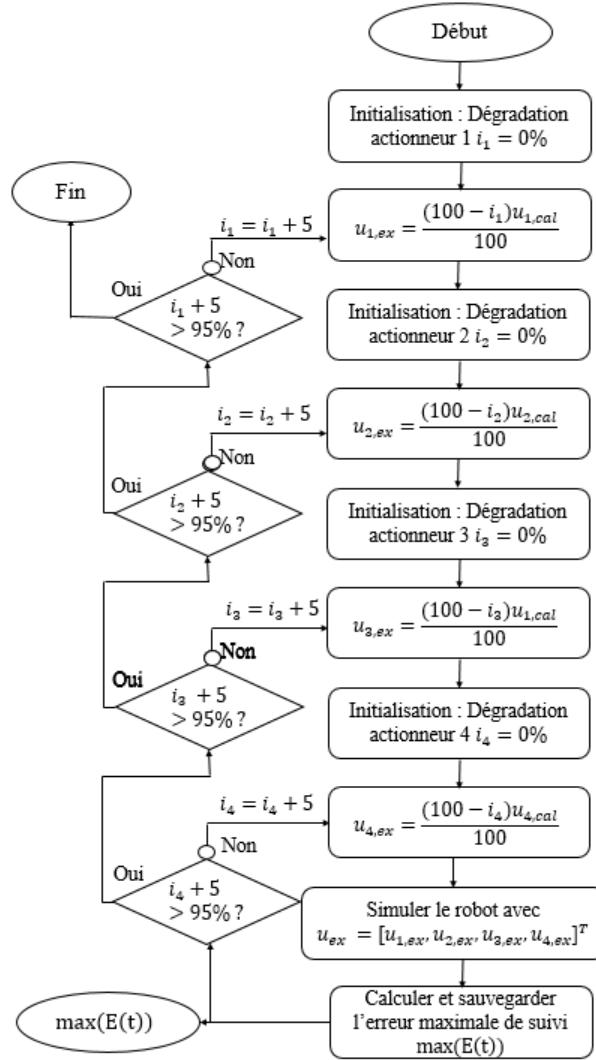


FIGURE III.21 – Simulation des dégradations des configurations.

Afin d'illustrer l'impact de ces dégradations sur le comportement du robot, prenons le **Scénario #1** suivant :

- **Scénario #1** : Atteindre la position désirée $pos_t = [150, 1, 0]^T$ en commençant par la position initiale $pos_0 = [0, 0, 0]^T$ et en passant par les deux positions suivantes $pos_1 = [50, 50, 0]^T$, $pos_2 = [90, 0, 0]^T$, voir la Figure III.2, Section III.3. Ce **Scénario #1** est exécuté avec et sans *défauts* suivant les trois cas d'études suivants :

- **Cas a** : Sans *défaut*.

- **Cas b** : Tous les actionneurs fonctionnent avec seulement 5 % de leur efficacité (*i.e.* ils ont tous perdu 95 % de leur efficacité).
- **Cas c** : Un actionneur est complètement défectueux, et les trois autres fonctionnent avec 5 % de leur efficacité.

La première partie de la Figure III.22, notée Figure III.22 (a) montre le suivi de trajectoire du robot pour ces trois cas d'études. Comme on peut le constater sur cette figure, le robot peut toujours fonctionner malgré ces *défaux* actionneurs importants, mais en mode dégradé. Lors des simulations, les erreurs maximales obtenues pour le **Scénario #1** sont comme suit : $E_m = 0,085 \text{ m}$ en mode nominal (**Cas a**), $E_m = 1,947 \text{ m}$ et $E_m = 2,367 \text{ m}$ dans le **Cas b** et **Cas c** respectivement.

Problème : Ces erreurs sont-elles acceptables dans la fab ? Le robot peut-il fonctionner sous ces modes dégradés dans l'usine ?

La réponse est non ! Car d'une part, un environnement collaboratif tel qu'une fab contient des opérateurs humains et des équipements. Cela restreint l'espace de travail du robot. Ainsi, les erreurs maximales de suivi E_m obtenues en mode dégradé (Figure III.22 (a)) sont trop importantes, et le comportement du robot peut présenter un réel danger pour les équipements et les personnes à proximité.

D'une autre part, travailler dans un environnement dynamique (*i.e.*, une fab) rend le fait de rencontrer des obstacles inattendus inévitable. Néanmoins, ces obstacles doivent être évités. Selon la stratégie d'évitement d'obstacles adaptée par le robot ainsi que la vitesse du contournement, les erreurs E_m obtenues précédemment peuvent être beaucoup plus importantes. Afin de montrer cela, le **Scénario 2** suivant est exécuté pour les trois mêmes cas étudiés dans le **Scénario #1**.

- **Scénario #2** : Exécuter le même **Scénario #1**, mais avec deux obstacles imprévus qui apparaissent sur le chemin du robot aux instants $t = 30 \text{ s}$ et $t = 150 \text{ s}$. Ces obstacles sont évités ou contournés par le robot avec une vitesse maximale de $V = 1.5 \text{ m/s}$.

Comme on peut le voir sur la Figure III.22 (b), où les erreurs de suivi suivantes sont obtenues pour les trois cas étudiés précédemment respectivement : $E_m = 0,308 \text{ m}$, $E_m = 3,861 \text{ m}$ et $E_m = 4,626 \text{ m.}$, les erreurs de suivi augmentent considérablement comparé aux cas sans obstacles. De ce fait, le comportement du robot devient plus dangereux.

III.6.3 Détermination des limites de sécurité

L'objectif principal de cette partie est de proposer une méthodologie permettant de définir des limites de dégradation à ne pas dépasser, tout en garantissant que le robot respecte les instructions de sécurité imposées par l'utilisateur, à savoir l'écart maximal autorisé par rapport à la trajectoire prédefinie. Cet écart maximal est défini comme un seuil associé à la limite de sécurité S_l . Pour cela, les erreurs maximales $E_m(P_1, P_2, P_3, P_4)$ enregistrées pour toutes les configurations de dégradation dans l'Algorithme donné par l'organigramme de la Figure III.23 pour les deux cas d'études (*i.e.* avec 4 et avec seulement 3 actionneurs) sont comparés à la valeur S_l définie par l'utilisateur, puis uniquement

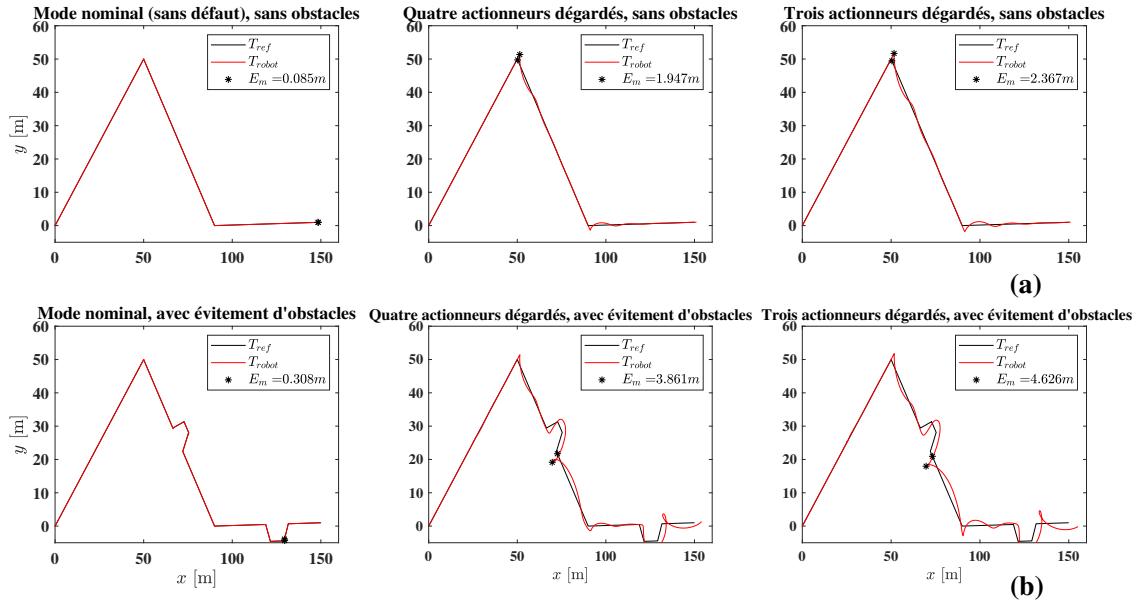


FIGURE III.22 – Suivi de trajectoire sous différentes conditions de fonctionnement.

les configurations de dégradation assurant une erreur inférieure à S_l sont acceptées, en suivant les étapes de l'organigramme de la Figure III.24.

III.6.4 Résultats de simulation

Les paramètres physiques du robot KUKA-KMP sont donnés dans la TABLE III.4, Section III.4.5. Le bruit de mesure est simulé en tenant compte de la précision des capteurs données dans la Section III.5.3 précédente.

Dans ce qui suit, nous présentons les résultats de simulation de quelques *défauts* simultanés au niveau des actionneurs, suivi de quelques résultats de la détermination des limites de sécurité pour des scénarios et conditions différents. À noter que sur les figures III.25, III.26, et III.27, les variables u_i , $i = 1, \dots, 4$ désignent les entrées calculées $u_{i,cal}$.

III.6.4.1 Résultats de simulation du FDI des *défauts* actionneurs simultanés

Dans la Figure III.25, les deux actionneurs A_3 et A_2 sont supposés perdre respectivement 5 % et 20% de leur efficacité. Ces *défauts* sont simulés à l'instant $t = 28.8 s$ et $t = 36 s$. Ils sont respectivement détectés et isolés 0.6 s et 1.8 s après leur apparition.

Des pertes d'efficacité avec un pourcentage de 5 %, 15 %, et 25 % sont simulés aux instants $t = 9.5 s$, $t = 28 s$, et $t = 50.2 s$ au niveau des actionneurs A_1 , A_2 et A_4 respectivement. Ces *défauts* sont détectés et isolés juste quelques secondes après apparition, voir Figure III.26.

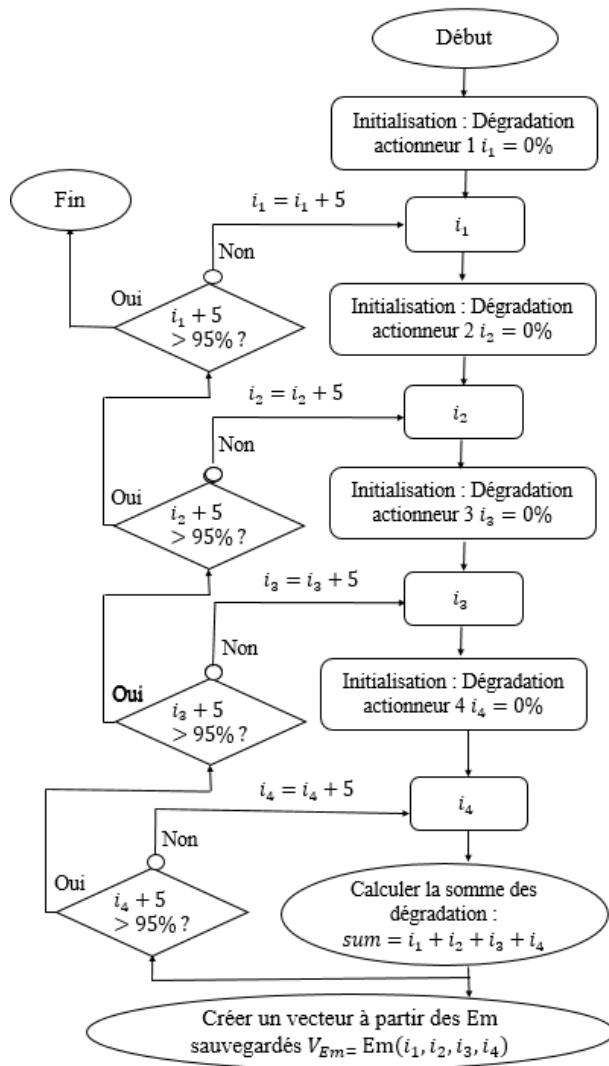


FIGURE III.23 – Sauvegarde des erreurs maximales de toutes les configurations dans un vecteur.

Enfin, des *défauts* représentants des pertes d'efficacité de 5 %, 10 %, 15 %, et 25 % sont simulés respectivement aux instants $t = 16.5 \text{ s}$, $t = 32 \text{ s}$, $t = 37.5 \text{ s}$, et $t = 37.9 \text{ s}$ au niveau des 4 actionneurs A_1, A_2, A_3 , et A_4 respectivement. Les résultats du FDI de ces *défauts* sont illustrés sur la Figure III.27.

III.6.4.2 Résultats de simulation de la détermination des limites de sécurité

Afin de déterminer les limites de dégradation pouvant être atteintes par les actionneurs sans que cela impacte considérablement le comportement du robot, et pour des raisons de simplification, la sommation du pourcentage de dégradation des actionneurs est adoptée dans ce travail. Cela permet d'éviter la complexité des calculs par rapport au cas du traitement individuel des dégradations de chaque actionneur, mais aussi d'obtenir un indicateur d'état de santé simple et facile à exploiter.

Dans les simulations, le seuil S_l correspondant à l'écart maximal toléré entre le robot et son chemin prédéfini dans la fab, est fixé à $S_l = 0.5 \text{ m}$, et ceci afin de respecter les conditions de fonctionnement

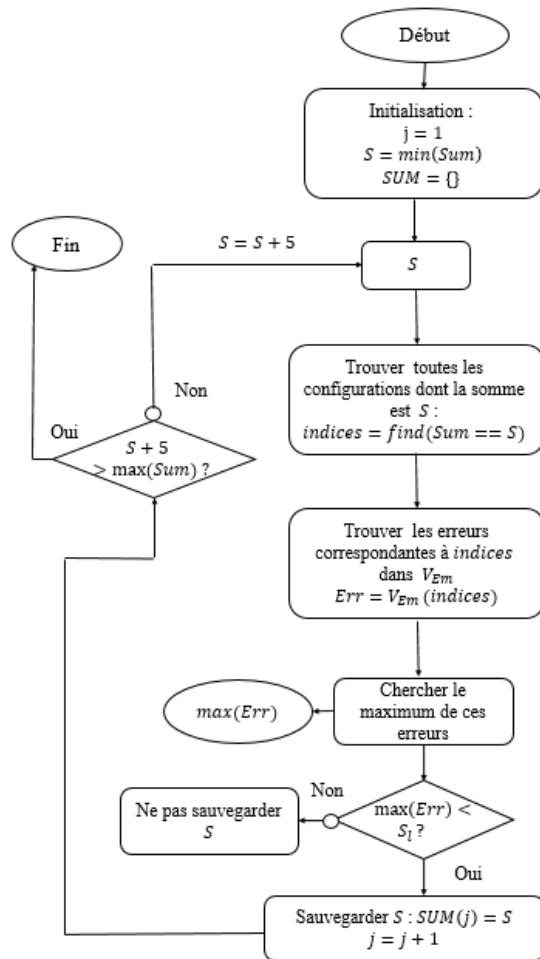


FIGURE III.24 – Détermination des limites des dégradations acceptées.

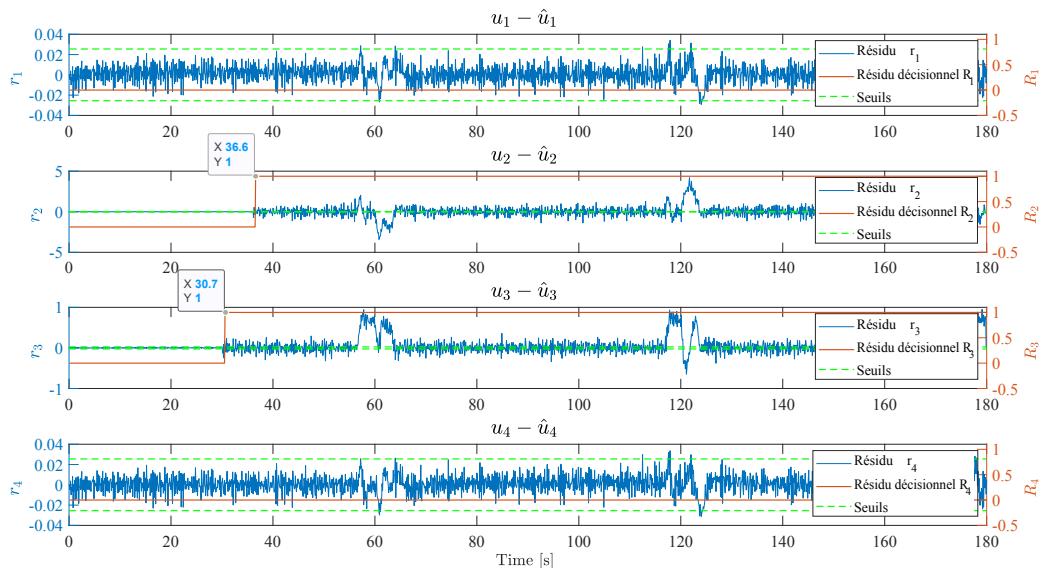


FIGURE III.25 – Les résidus en présence d'un défaut au niveau de l'actionneur de la roue 2 et 3.

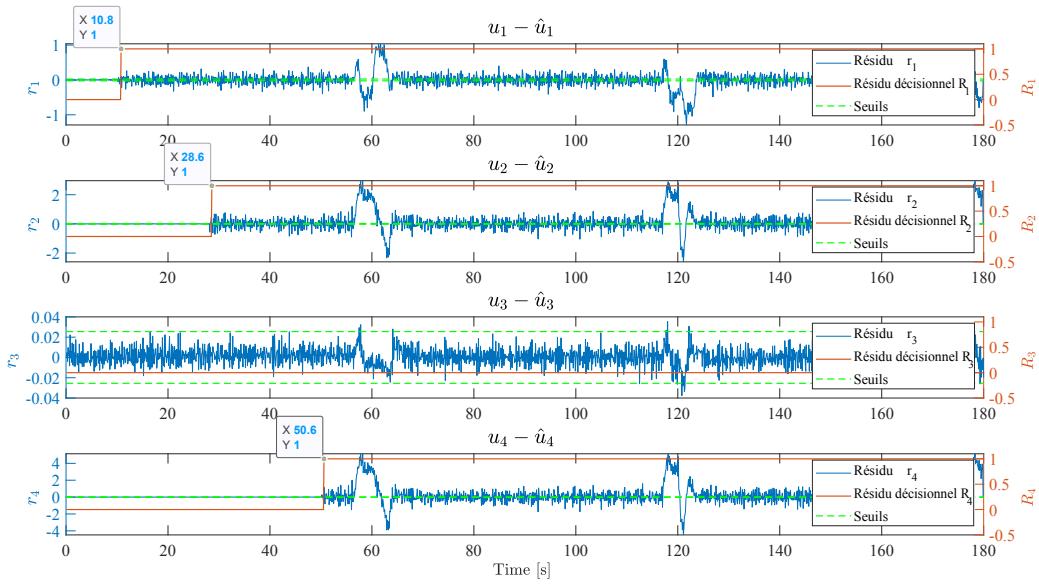


FIGURE III.26 – Les résidus en présence d'un défaut au niveau de l'actionneur de la roue 1, 2 et 4.

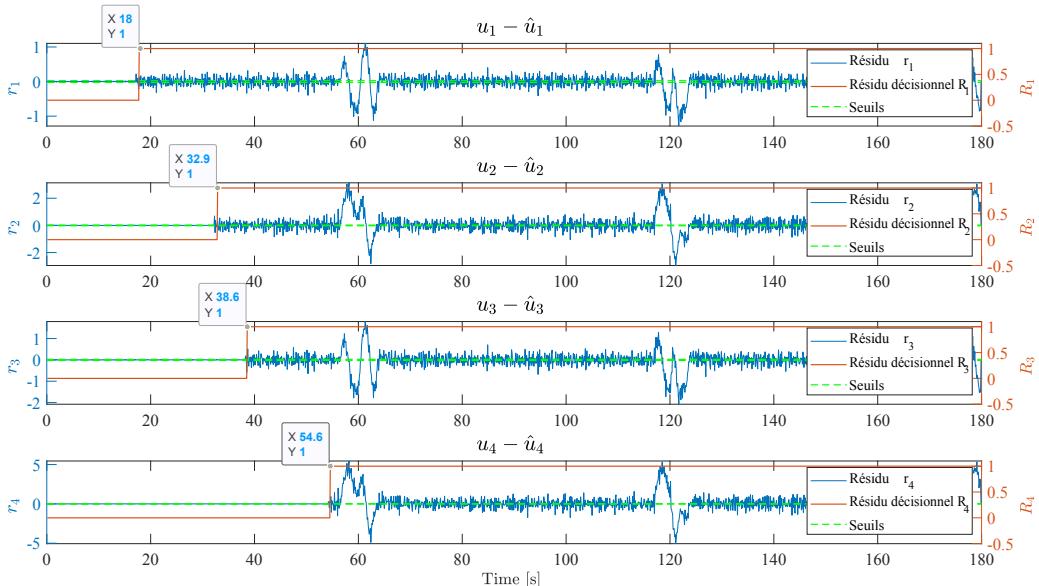


FIGURE III.27 – Les résidus en présence d'un défaut au niveau des actionneurs des quatre roues.

dans la fab. Des simulations sont faites en l'absence et en présence d'obstacles mobiles pour les deux cas d'étude (**Cas 1** et **Cas2**) cités précédemment dans la Section III.6. Différents scénarios sous différentes conditions de fonctionnement sont testés afin d'étudier l'impact des missions du robot et des conditions de fonctionnement sur le comportement du robot en mode dégradé.

Supposons que le robot doit exécuter le **Scénario #1** sans obstacles, puis le **Scénario #2** avec deux obstacles imprévus sur son chemin. Ces deux scénarios sont donnés précédemment dans la Section III.6.2. Ils sont simulés pour les deux cas d'études : 1) avec 4 actionneurs, et 2) avec 3 actionneurs seulement, le 4^{ème} est supposé être complètement défaillant).

1) Cas 1 avec 4 actionneurs. La Figure III.28 montre l'évolution de E_m en fonction de la somme des pourcentages des dégradations des actionneurs, comparé au seuil $S_l = 0.5 \text{ m}$. Cela a pour but de déterminer les limites de dégradation que le robot peut atteindre tout en garantissant que son comportement ne présente aucun danger pour ce qu'il y a autour de lui.

Comme montré sur cette Figure III.28, les dégradations peuvent atteindre un pourcentage de 165% au niveau des 4 actionneurs en absence d'obstacles, tandis qu'elles ne peuvent pas dépasser le pourcentage de 100 % dans le cas d'obstacles contournés avec une vitesse de $V = 1.5 \text{ m/s}$.

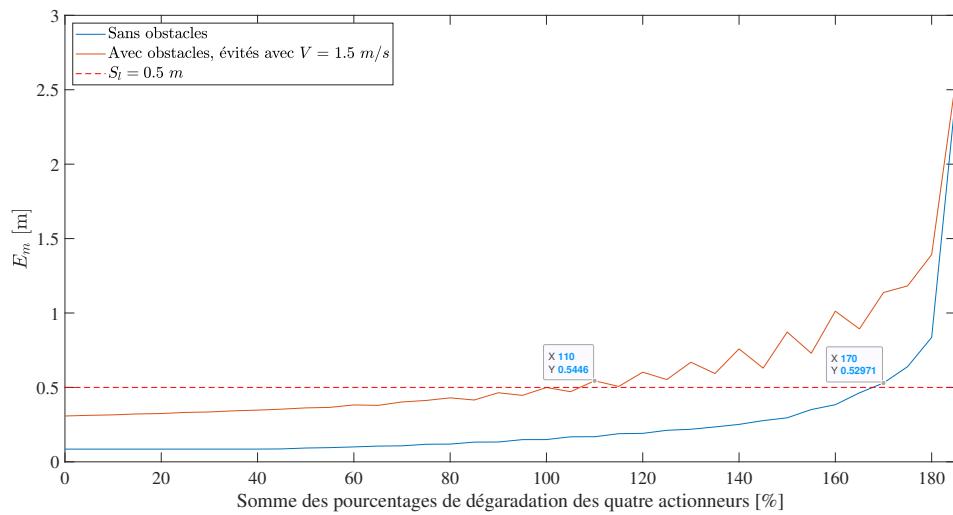


FIGURE III.28 – Limites des dégradations acceptées dans le cas avec quatre actionneurs en absence et en présence d'obstacles.

2) Cas 2 avec seulement 3 actionneurs. Les mêmes scénarios : **Scenario #1** sans obstacles, et **Scenario #2** avec obstacles sont simulés pour le cas avec seulement trois actionneurs. Dans ce cas de figures, toutes les configurations par rapport à l'actionneur complètement défaillant ont été testées. Les limites des dégradations tolérées sont illustrées dans la Figure III.29. Ces limites sont d'environ 84 % dans le cas sans obstacles, et seulement d'environ 30 % pour le cas avec des obstacles évités ou contournés à une vitesse de $V = 1.5 \text{ m/s}$.

III.6.5 Amélioration des limites de sécurité

Les limites de dégradation tolérées données dans la Section III.6.4.2 précédente sont calculées en respectant les conditions initiales de fonctionnement, fixées par l'utilisateur pour exécuter les missions en mode nominal. Après une détection d'un *défaut*, et afin de réduire son impact sur le comportement du robot, l'utilisateur peut prendre en compte certaines stratégies afin d'améliorer les limites de dégradation tolérées, parmi lesquelles on peut suggérer :

- 1) Changer la trajectoire ou le chemin de la mission du robot (*i.e.*, pour la même mission, définir un chemin plus lisse).

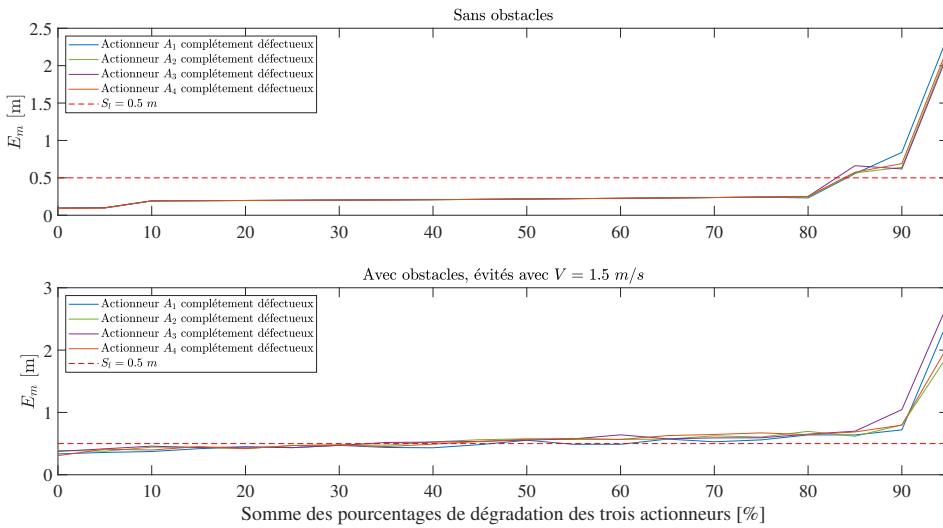


FIGURE III.29 – Limites des dégradations acceptées dans le cas avec trois actionneurs en absence et en présence d'obstacles.

- 2) Réduire la vitesse du robot entre les positions intermédiaires (e.g. avant d'atteindre pos_1 ou pos_2 dans le **Scénario #1**).
- 3) D'autres stratégies peuvent être prises en compte comme par exemple réduire le nombre de FOUPs transportés par le robot, ou tenir les opérateurs de la fab informés sur le mode dégradé du robot afin qu'ils évitent de laisser des obstacles mobiles (e.g. chaises, chariots, ...) sur le chemin du robot par exemple.

Ces stratégies proposées restent valables que ce soit pour le cas sans obstacles imprévus, ou pour le cas avec obstacles. Dans ce travail, nous étudions uniquement les deux premières stratégies. L'efficacité de la première stratégie est présentée pour le cas sans obstacles. Cela se fait en modifiant le chemin du robot, en prenant un chemin plus lisse. Quant à l'efficacité de la deuxième stratégie, elle est illustrée pour le cas avec obstacles, en utilisant une vitesse d'évitement d'obstacles de $V = 1 \text{ m/s}$ au lieu de $V = 1,5 \text{ m/s}$. À noter que le contrôle de cette vitesse se fait par saturation (en supposant que l'actionneur ne peut pas dépasser $V = 1 \text{ m/s}$ ou $V = 1.5 \text{ m/s}$) pendant l'évitement des obstacles.

Afin d'illustrer la première stratégie, le **Scénario #3** suivant est simulé avec quatre, puis avec seulement trois actionneurs.

- **Scénario #3 :** On suppose que le robot doit exécuter le **Scénario #1**, mais au lieu d'aller jusqu'à $y = 50 \text{ m}$ de pos_1 , il passe par $pos'_1 = (50, 15, 0)$. Cela permet d'avoir un chemin plus lisse comme on peut le voir sur la Figure III.30, où les trajectoires $y(x)$ des deux scénarios **Scénario #1** et **Scénario #3** sont illustrées pour comparaison.

Les résultats des limites de sécurité obtenus pour cette première stratégie sont donnés dans la Figure III.31 pour les deux cas d'études : avec quatre et avec seulement trois actionneurs. Comme on peut le voir sur cette Figure III.31, et comparé aux résultats obtenus pour le **Scénario #1** sans obstacles, les dégradations peuvent atteindre 180 % avec quatre actionneurs, et environ 90 % avec

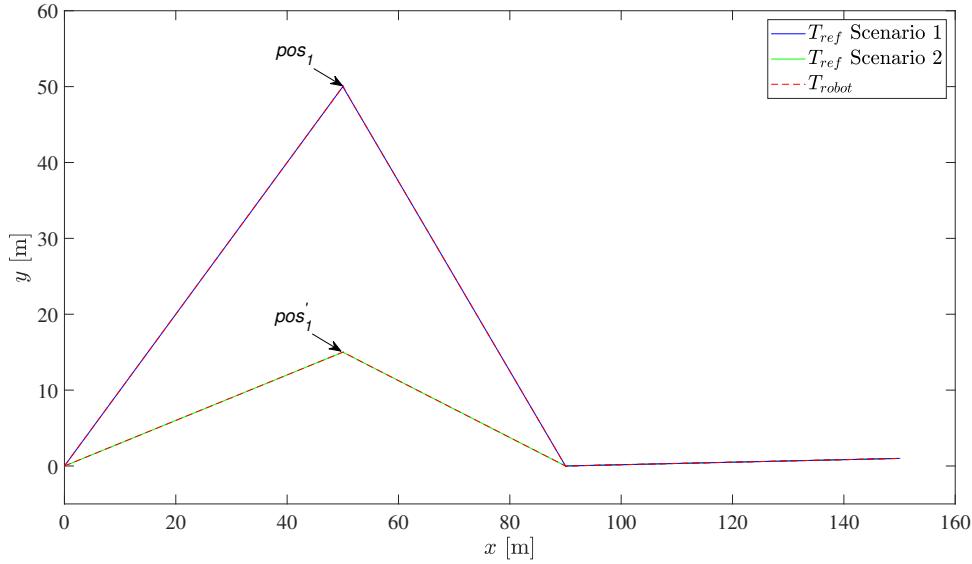


FIGURE III.30 – Scénario #3 comparé au Scénario #1 en absence d’obstacles.

trois actionneurs. Avant de changer le chemin du robot, ces limites étaient respectivement à 167 % et 84 %. À noter que plus l’angle formé entre deux positions intermédiaires est obtus, plus les limites de sécurité sont importantes.

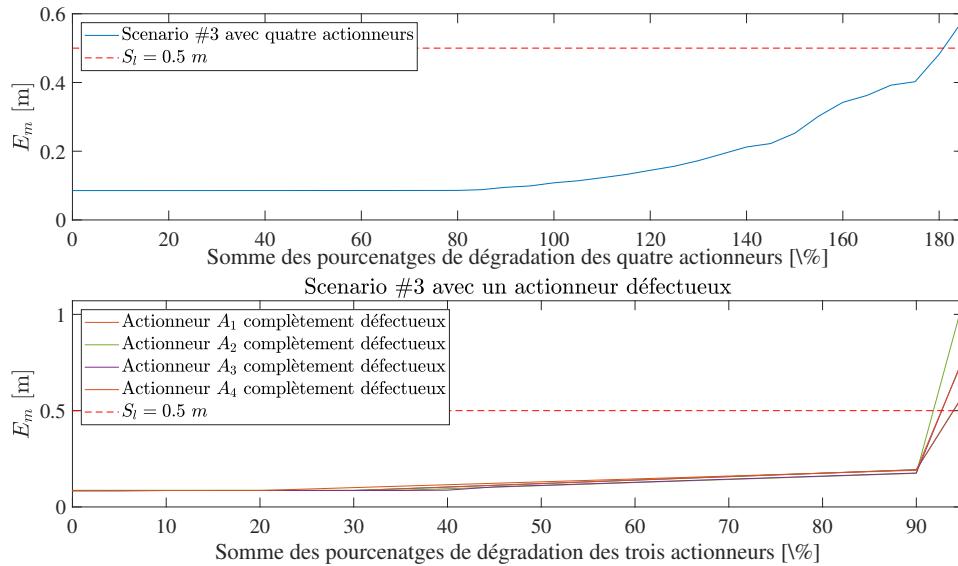


FIGURE III.31 – Limites de dégradation obtenues pour Scénario #3 pour les deux cas d’études : avec quatre et avec trois actionneurs.

Quant à la deuxième stratégie proposée, la Figure III.32 montre les limites de dégradation tolérées obtenues pour les deux cas : avec 4 et avec seulement 3 actionneurs, et ceci en évitant les obstacles du Scénario #2 avec une vitesse de $V = 1 \text{ m/s}$ au lieu de $1,5 \text{ m/s}$. Comme on peut le voir dans ces résultats, les limites de dégradation tolérées sont beaucoup plus élevées avec l’utilisation d’une faible

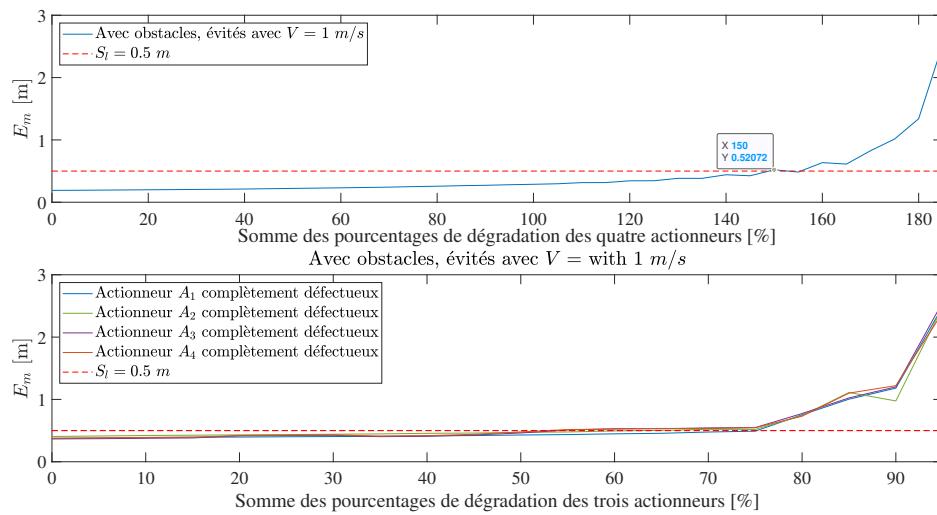


FIGURE III.32 – Limites de dégradation obtenues pour **Scénario #2** en évitant les obstacles avec $V = 1 \text{ m/s}$ pour les deux cas d'études : avec quatre et avec trois actionneurs.

vitesse. Ces dernières sont d'environ 148 % dans le premier cas contre 100 % obtenus avec $V = 1,5 \text{ m/s}$, et d'environ 55 % dans le second cas contre seulement 30 % obtenus avec $V = 1,5 \text{ m/s}$ (voir Figure III.29).

III.6.6 Discussion

Il est important de noter que les limites de sécurité présentées dans ce travail sont obtenues en sauvegardant uniquement les configurations de dégradation qui donnent une déviation maximale par rapport à la trajectoire de référence pour chaque somme de pourcentages (voir l'organigramme de la Figure III.21). Pour le même niveau de dégradation, et en fonction de la mission du robot, il peut y avoir des configurations où $E_m < S_l$, pour lesquelles l'arrêt du robot n'est pas nécessaire. Afin d'illustrer cela, quelques exemples de configurations avec les déviations maximales E_m correspondantes sont données pour le même niveau de dégradation dans la TABLE III.8 suivante :

TABLE III.8 – Quelques configurations de dégradations pour le même niveau de dégradation.

Niveau de dégradation	Configuration	$E_m [\text{m}]$
Sans obstacles Dégradation totale = 170%	[85 0 85 0]	$0.5297 > S_l$
	[80 5 85 0]	$0.4077 < S_l$
	[75 15 75 5]	$0.3782 << S_l$
Avec obstacles, contournement avec $V = 1.5 \text{ m/s}$ Dégradation totale = 110%	[0 55 0 55]	$0.5446 > S_l$
	[45 10 50 5]	$0.42815 < S_l$
	[90 15 0 5]	$0.3671 << S_l$
Avec obstacles, contournement avec $V = 1 \text{ m/s}$ Dégradation totale = 150%	[0 75 0 75]	$0.5207 > S_l$
	[5 75 0 70]	$0.4075 < S_l$
	[70 15 65 0]	$0.3550 << S_l$

Ici, seuls les résultats pour le premier cas d'étude (*i.e.* avec quatre actionneurs) et pour le **scénario #1** en absence d'obstacles, puis pour le **scénario #2** en présence des deux obstacles imprévus, contournés avec un vitesse de $V = 1,5 \text{ m/s}$ puis $V = 1 \text{ m/s}$ sont présentés à titre d'illustration, mais la remarque reste valable pour le deuxième cas d'étude (*i.e.* un actionneur est complètement défaillant) et cela quel que soit le scénario.

Les configurations des dégradations sont données dans l'ordre suivant : $[P_1\%, P_2\%, P_3\%, P_4\%]$ (voir Figure III.1, Section III.2 pour la numérotation des actionneurs). Les colonnes rouges dans la TABLE III.8 indiquent les limites de dégradations non tolérées, où S_l est légèrement dépassé, comme on peut le voir respectivement sur la Figure III.28 et la Figure III.32 pour le cas avec quatre actionneurs. Comme le montre cette TABLE III.8, et pour le même niveau de dégradation (*i.e.* même somme des pourcentages de perte d'efficacité des actionneurs), il existe des configurations où les limites de sécurité définies précédemment peuvent être dépassées sans poser de problème par rapport au comportement du robot : $(E_m < S_l)$.

III.7 Conclusion

Nous avons développé dans ce **Chapitre III** deux approches de diagnostic. La première pour détecter et isoler à la fois les *défaux* actionneurs et capteurs des robots omnidirectionnels à 4 roues Mecanum (4-MWMR). La méthode est basée sur l'utilisation d'un observateur de Kalman étendu (EKO) pour générer des résidus. Comme le montre les résultats de cette approche, les *défaux* capteurs sont détectés et isolés en se basant uniquement sur la table de signature des résidus, tandis que pour le cas des *défaux* actionneurs nous avons besoin de plus de caractéristiques sur les résidus (*i.e.* rapidité et pourcentage de détection) afin d'être isolés.

La deuxième approche est une méthode permettant l'isolation des *défaux* actionneurs d'une façon plus simple. Elle est basée sur l'utilisation d'un banc d'observateurs à entrées inconnues (UIOs) pour les systèmes à paramètres variants (LPVS). Les résidus générées avec cette deuxième approche ont une signature caractéristique de chaque *défaut* actionneur. Par conséquent, ces derniers sont isolés juste en considérant la table de signature des résidus.

Après cette étape de diagnostic des *défaux*, nous nous sommes intéressés à une autre problématique concernant l'étude de l'impact des *défaux* actionneurs sur le comportement des robots. Nous nous sommes focalisés spécialement sur les *défaux* actionneurs car ces derniers sont plus susceptibles d'apparaître suite à des sollicitations importantes des robots dans la fab, et qui accélère le vieillissement. Une approche visant à surveiller l'évolution de l'état de santé des actionneurs après une première détection d'un *défaut*, et à étudier l'effet de cet état de santé sur le comportement des robots est proposée. Le but derrière cette étude est de déterminer des limites de sécurité que les actionneurs peuvent atteindre au niveau des dégradations tout en garantissant que le comportement des robots reste acceptable dans la fab. Ceci permet de fournir une aide à la décision (Decision Making Support)

quant à la nécessité d'intervention humaine ou pas, et permet d'éviter l'arrêt immédiat des robots après détection d'un *défaut* quand cela n'est pas nécessaire.

CHAPITRE IV

RECONFIGURATION DE LA TRAJECTOIRE ET COMPENSATION DU RETARD GÉNÉRÉ PAR LES OBSTACLES MOBILES

IV.1	Introduction	123
IV.2	Planification de missions et génération de trajectoire	124
IV.2.1	Planification des missions et du chemin (Missions and path planning)	124
IV.2.2	Génération de trajectoire	125
IV.3	Suivi de trajectoire et évitemen t d'obstacles	128
IV.3.1	Application aux robots omnidirectionnels à 4 roues Mecanum (4-MWMRs)	128
IV.3.2	Évitement d'obstacles	129
IV.4	Reconfiguration de la trajectoire et gestion du retard	131
IV.4.1	Le principe de la gestion du retard	131
IV.4.2	Analyse de la compensation du retard	133
IV.5	Résultats de simulation	134
IV.5.1	Les scénarios	134
IV.5.2	Simulations	135
IV.5.3	Vérification et analyse de la possibilité de compensation du retard	137
IV.6	Conclusion	138

IV.1 Introduction

Comme évoqué dans l'**Introduction Générale**, la fabrication d'un dispositif semi-conducteur passe par un nombre important d'étapes de production (environ 600) entre les différents équipements de la fab. Chaque équipement réalise un traitement spécifique et indispensable afin d'avoir un produit fini de bonne qualité.

La vie d'un lot de wafers dans la fab peut être considérée comme étant une répétition de trois états : 1) en attente, 2) en traitement, et 3) en transport. Les états d'attente et de transport sont considérés comme étant des temps de non-production car pendant ces deux états, les wafers ne subissent aucun traitement particulier. L'attente et le transport n'apportent pas de valeur ajoutée [10] aux lots.

Même si l'état en transport n'apporte aucun changement implicite sur l'état des wafers, néanmoins il y contribue car il permet la continuité de l'enchaînement des étapes de production au niveau des équipements. Cela permet une optimisation de la productivité des équipements, ce qui améliore la productivité globale.

La fab est un environnement dynamique. Elle contient des obstacles fixes mais aussi des obstacles mobiles tels que les opérateurs humains par exemple. Cela restreint l'espace de fonctionnement des robots et rend la rencontre des obstacles imprévisibles sur le chemin des robots durant la navigation inévitable. Quand le robot rencontre un obstacle qui l'empêche de poursuivre sa mission, il a deux choix possibles afin d'éviter une collision :

1. Contourner (éviter) l'obstacle s'il y a assez d'espace autour.
2. S'arrêter et attendre que l'obstacle s'éloigne de son chemin si le contournement n'est pas possible.

Quel que soit le comportement adapté par le robot afin d'éviter la collision, rencontrer un obstacle imprévu sur son chemin génère du retard par rapport à la durée planifiée de la mission en cours.

Cependant, pour certaines étapes de production, et afin que la production soit optimale, les lots doivent tous arriver à temps au niveau de l'équipement avant que celui-ci ne soit mis en marche pour le traitement d'une nouvelle série de lots (sachant qu'il y a des étapes de traitement qui peuvent durer longtemps). Tout retard au niveau du dépôt des lots entraîne :

1. Soit la mise en marche de l'équipement sans attendre que tous les lots soient arrivés. Cela impacte le rendement de la production.
2. Soit un retard sur la productivité de l'équipement si ce dernier attend l'arrivée de tous les lots pour commencer le traitement. Cela engendre un retard dans la chaîne de production.

Afin d'éviter ou tout au moins de minimiser l'impact des obstacles mobiles sur l'efficacité des robots dans la fab, nous nous intéressons dans ce chapitre à la problématique de la compensation ou de la minimisation du retard généré par les obstacles mobiles.

Ce chapitre est consacré à la description des travaux réalisés pendant cette thèse afin de proposer une méthodologie permettant de respecter au mieux la durée planifiée pour chaque mission de transport d'un lot [1, 3, 9], en minimisant au mieux le retard généré par les obstacles mobiles. Dans ces travaux, cette méthodologie est appliquée sur les robots omnidirectionnels 4-MWMRs afin de montrer son efficacité mais elle reste valable pour les robots unicycles et les autres types de robots mobiles.

Ce chapitre commence par la description de la méthode de navigation adoptée dans ces travaux, allant de la planification du chemin à la génération d'une trajectoire conforme à ce chemin afin de permettre au robot de réaliser une mission de transport entre deux équipements situés dans deux endroits différents dans la fab, voir Section IV.2.

Ensuite, nous décrivons l'approche que nous avons utilisée dans les simulations pour l'évitement d'obstacles dans la Section IV.3. Après cela, nous détaillons dans la Section IV.4 la méthodologie que nous avons proposée afin de minimiser ou de compenser complètement (quand c'est possible) le retard que l'on peut avoir sur la réalisation de la mission par les robots, ce retard étant lié aux obstacles imprévus sur le chemin du robot. Enfin, et afin de valider cette approche, nous exposons quelques résultats de simulation pour différents scénarios simulés dans la Section IV.5.

IV2 Planification de missions et génération de trajectoire

Dans ce travail, afin d'être le plus proche possible de la réalité des cas d'étude en tenant compte des conditions de l'environnement réel du fonctionnement des robots considérés (*i.e.* la fab), ainsi que des cas d'utilisation des robots (Robots use-cases in the fab), nous utilisons une approche de navigation hybride (voir Section I.6).

D'une manière générale, les approches de navigation hybrides consistent à planifier un chemin global (Global path planning) en se basant sur la carte de l'environnement (*i.e.* en utilisant une approche délibérative), puis à éviter les obstacles mobiles et imprévus lors de la navigation, en se basant sur les données disponibles des capteurs de navigation (*i.e.* les télémètres, les caméras, ...) embarqués sur les robots (en utilisant une approche réactive) [136].

Dans ce travail, les robots dans la fab doivent réaliser des missions de transport de produits d'une unité d'équipement située à une position initiale, notée p_0 , à une autre unité d'équipement située dans une position désirée, notée p_f , en utilisant une approche hybride.

IV2.1 Planification des missions et du chemin (Missions and path planning)

Planifier une mission m pour les robots dans la fab consiste à donner :

1. La position initiale et la position finale (désirée).
2. La durée de la mission.
3. Le chemin à suivre afin d'atteindre la destination finale souhaitée.

Concernant le problème de planification du chemin, et en raison des conditions de travail complexes dans la fab, à savoir le manque d'espace, le robot doit passer par des positions intermédiaires (*i.e.* il doit exécuter des sous-missions m_i) avant d'arriver à la destination désirée d'une mission globale m .

Ces positions intermédiaires sont atteintes en suivant des lignes droites, planifiées en se basant sur la carte de l'environnement du travail du robot. Quant au problème de la définition de la durée de la mission t , la durée est fixée en fonction des exigences du processus de fabrication, tout en tenant compte de la durée requise t_i pour chaque sous-mission m_i .

Afin de ne pas impacter le rendement de la production, le robot doit exécuter les sous-missions m_i en respectant au mieux la durée t_i planifiée pour chaque sous-destination. Dans ce cas, la durée t de la mission globale m sera respectée. Donc pour chaque sous-mission m_i , générer des trajectoires prenant en compte la notion du temps et les chemins prédéfinis est important.

IV.2.2 Génération de trajectoire

Dans la littérature, il existe plusieurs méthodes de génération de trajectoire comme par exemple à base de courbes de Bezier [201] ou en utilisant les splines [202]. La génération de trajectoire n'est pas le propos de cette thèse. Nous n'allons donc pas nous focaliser sur la génération de trajectoire (nous ne cherchons pas à l'optimiser par exemple). Le but ici est de proposer une façon industrialisable de générer la trajectoire. La méthode proposée repose sur la mise en place de trajectoires en vitesse sous forme de trapèze comme montré sur la Figure IV.1.

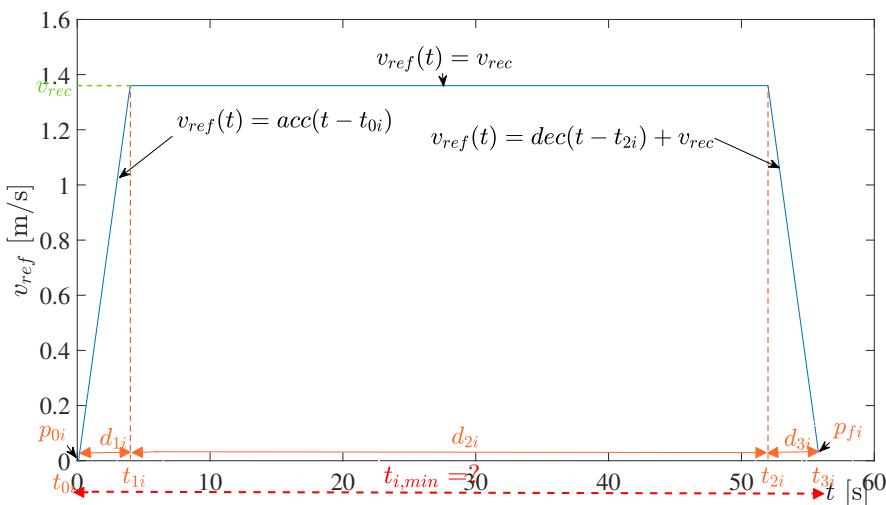


FIGURE IV.1 – Principe de calcul de $t_{i,min}$ pour la vérification de la faisabilité d'une sous-mission m_i .

Il est à noter que cette méthode peut ne pas être optimale, néanmoins elle est suffisante pour présenter et démontrer notre méthodologie de compensation du retard. Cette méthodologie reste valable quelle que soit la méthode de génération de trajectoire utilisée.

Afin de faciliter la collaboration entre les opérateurs humains et les robots mobiles dans la fab, une valeur de vitesse recommandée, notée v_{rec} , est définie par l'utilisateur. Cette vitesse ne doit pas être dépassée par les robots dans les conditions normales de fonctionnement. Cette valeur recommandée est inférieure à la vitesse maximale v_{max} que les robots peuvent atteindre dans la fab, et qui garantie le respect des consignes de sécurité et assure que le comportement des robots ne présente aucun risque que ce soit pour les robots eux-mêmes ou pour les personnes à proximité.

Avant de générer une trajectoire de référence de vitesse $v_{ref}(t)$ pour une sous-mission m_i , la faisabilité de cette sous-mission m_i par rapport à sa durée planifiée t_i doit être vérifiée. Pour cela, t_i doit être supérieur à $t_{i,min}$, le temps minimal nécessaire pour exécuter m_i tout en fonctionnant avec la vitesse recommandée v_{rec} dans la fab.

Ce $t_{i,min}$ est calculé en prenant en compte v_{rec} , la position du départ et de la position finale désirée de la sous-mission m_i , $p_{0i} = [x_{0i}; y_{0i}; \theta_{0i}]^T$, et $p_{fi} = [x_{fi}; y_{fi}; \theta_{fi}]^T$, ainsi que les valeurs d'accélération et de décélération du robot, notées acc et dec respectivement. Pour des raisons de simplification, la valeur de décélération est considérée comme $|dec| = |acc|$. $t_{i,min}$ est calculé en suivant les étapes suivantes (voir la Figure IV.1 pour illustration) :

1. **Calculer la distance totale d_i à parcourir entre p_{0i} et p_{fi} :** $d_i = \sqrt{(x_{fi} - x_{0i})^2 + (y_{fi} - y_{0i})^2}$.
2. **Calculer Δt_{1i} , le temps nécessaire afin d'atteindre $v_{ref}(t) = v_{rec}$:** $\Delta t_{1i} = \frac{v_{rec}}{acc}$.
3. **Déduire la distance Δd_{1i} à parcourir durant Δt_{1i} :** $\Delta d_{1i} = v_{rec} \Delta t_{1i}$.
4. **Prenant en compte $|acc| = |dec|$, on a :** $\Delta t_{3i} = \Delta t_{1i}$ et $\Delta d_{3i} = \Delta d_{1i}$.
5. **Calculer la distance Δd_{2i} à parcourir durant Δt_{2i} :** $\Delta d_{2i} = d_i - 2\Delta d_{1i}$.
6. **Déduire Δt_{2i} :** $\Delta t_{2i} = \frac{\Delta d_{2i}}{v_{rec}}$.
7. **Enfin, déduire $t_{i,min}$:**
$$t_{i,min} = 2\Delta t_{1i} + \Delta t_{2i}$$

Pour être sûr que les sous-missions m_i n'induisent aucun retard dans les conditions normales de fonctionnement (*i.e.* en absence d'obstacles mobiles), t_i doit être tel que $t_i \geq t_{i,min}$. Ici, deux cas sont distingués :

1. $t_i = t_{i,min}$.
2. $t_i > t_{i,min}$.

Dans le premier cas, la vitesse constante $v_{ref,i}$ pendant la période Δt_{2i} (voir la Figure IV.2 (a)) est $v_{ref,i} = v_{rec}$, alors que dans le second cas, $v_{ref,i}$ doit être calculée pour respecter la durée planifiée t_i . Cela se fait en suivant les étapes illustrées sur la Figure IV.2 (a) comme suit :

1. **Calculer la distance à parcourir entre p_{0i} et p_{fi} :** $d_i = \sqrt{(x_{fi} - x_{0i})^2 + (y_{fi} - y_{0i})^2}$.
2. **Calculer $v_{ref,i}$ pour exécuter la mission tout en respectant t_i en prenant la plus petite valeur obtenue en solvant l'équation suivante :** $\frac{1}{acc} v_{ref,i}^2 - t_i v_{ref,i} + d_i = 0$.
3. **Connaissant $v_{ref,i}$, calculer Δt_{1i} , le temps nécessaire pour atteindre $v_{ref,i}$ en commençant par t_{0i} :** $\Delta t_{1i} = \frac{v_{ref,i}}{acc}$.

4. A partir de l'hypothèse $|dec| = |acc|$, on a : $\Delta t_{3i} = \Delta t_{1i}$.

5. Déduire Δt_{2i} comme suit : $\Delta t_{2i} = t_i - 2\Delta t_{1i}$.

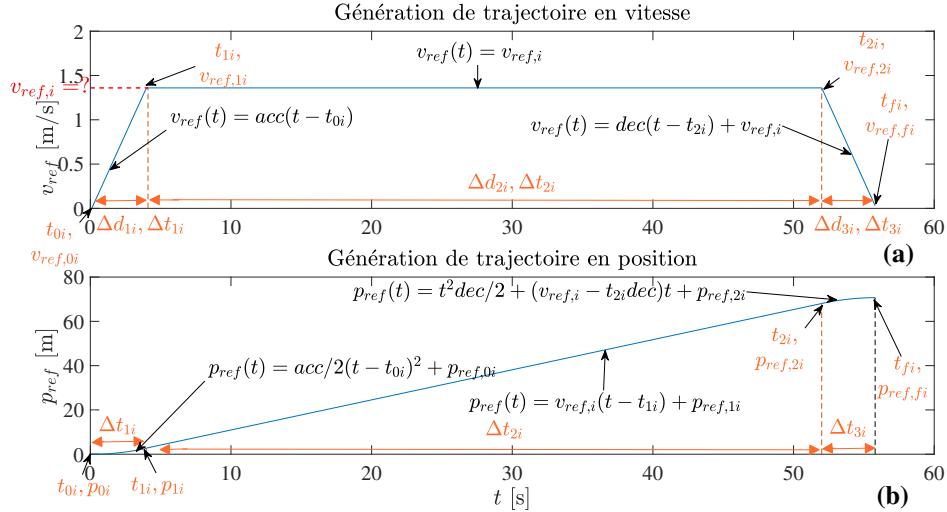


FIGURE IV.2 – Principe de génération des trajectoires de vitesse et de position.

Enfin, connaissant t_{0i} , t_{fi} , t_i , et $v_{ref,i}$, le profil de trajectoire de référence de vitesse $v_{ref}(t)$ est généré pour les deux cas cités précédemment comme suit (voir Figure IV.2 (a)) :

1. Déduire : $t_{1i} = \Delta t_{1i} + t_{0i}$ et $t_{2i} = \Delta t_{2i} + t_{1i}$.

2. Déduire le profile de vitesse :

$$v_{ref}(t) = \begin{cases} acc(t - t_{0i}) + v_{ref,0i} & \text{si } t = t_{0i} \rightarrow t_{1i} \\ v_{ref,i} & \text{si } t = t_{1i} \rightarrow t_{2i} \\ dec(t - t_{2i}) + v_{ref,i} & \text{si } t = t_{2i} \rightarrow t_{3i} \end{cases}$$

Quant à la trajectoire de référence associée à la position du robot $p_{ref}(t)$, elle est obtenue à partir du profile de trajectoire de vitesse $v_{ref}(t)$ par intégration temporelle comme suit (voir Figure IV.2 (b)) :

$$p_{ref}(t) = \begin{cases} \frac{acc}{2}(t - t_{0i})^2 + v_{ref,i}(t - t_{0i}) + p_{ref,0i} & \text{si } t = t_{0i} \rightarrow t_{1i} \\ v_{ref,i}(t - t_{1i}) + p_{ref,1i} & \text{si } t = t_{1i} \rightarrow t_{2i} \\ \frac{dec}{2}t^2 + (v_{ref,i} - t_{2i} dec)t + p_{ref,2i} & \text{si } t = t_{2i} \rightarrow t_{3i} \end{cases}$$

Enfin, pour avoir les profiles de trajectoire de vitesse et de position par rapport à l'axe \vec{x} et à l'axe \vec{y} , $v_{ref}(t)$ et $p_{ref}(t)$ sont projetés en respectant l'angle d'orientation θ obtenu à partir des mesures du gyroscope comme suit : $v_{x,ref}(t) = v_{ref}(t)\cos\theta$, $v_{y,ref}(t) = v_{ref}(t)\sin\theta$, $x_{ref}(t) = p_{ref}(t)\cos\theta$, et $y_{ref}(t) = p_{ref}(t)\sin\theta$.

IV.3 Suivi de trajectoire et évitement d'obstacles

IV.3.1 Application aux robots omnidirectionnels à 4 roues Mecanum (4-MWMRs)

Les trajectoires générées dans la section précédente sont appliquées aux robots omnidirectionnels à 4 roues Mecanum (4-MWMRs) en utilisant la loi de commande par placement de pôles donnée précédemment dans la Section III.3 par (III.10), avec :

- $Y_{ref}^0 = Y_{ref}$, la trajectoire de référence en position donnée par : $Y_{ref} = [x_{ref}(t); y_{ref}(t); \theta_{ref}(t)]^T$.
- $Y_{ref}^1 = \dot{Y}_{ref}$, la trajectoire de référence en vitesse donnée par : $\dot{Y}_{ref} = [\dot{x}_{ref}(t); \dot{y}_{ref}(t); \dot{\theta}_{ref}(t)]^T = [v_{x,ref}(t); v_{y,ref}(t); v_{\theta,ref}(t)]^T$.
- $Y_{ref}^2 = \ddot{Y}_{ref}$, la trajectoire de référence en accélération donnée par : $\ddot{Y}_{ref} = [\ddot{x}_{ref}(t); \ddot{y}_{ref}(t); \ddot{\theta}_{ref}(t)]^T = [a_{x,ref}(t); a_{y,ref}(t); a_{\theta,ref}(t)]^T$.

Grâce à la propriété de l'omnidirectionnalité, les 4-MWMRs sont capables d'atteindre n'importe quelle position dans leur environnement d'évolution sans avoir besoin de changer leur angle d'orientation θ . Ainsi, $\theta_{ref}(t)$ est considéré comme une constante par sous-mission. Par conséquent, $v_{\theta,ref}(t) = a_{\theta,ref}(t) = 0$. Quant à $a_{x,ref}(t)$ et $a_{y,ref}(t)$, ils sont déduits de la dérivation temporelle des trajectoires de vitesse $v_{x,ref}(t)$ et $v_{y,ref}(t)$ respectivement.

Afin de montrer l'efficacité de ce contrôle dans le suivi des trajectoires générées avec l'approche proposée, prenons l'exemple suivant : supposons que le robot doit atteindre la position $p_f = [100; 10; 0]^T$ en partant de la position initiale $p_0 = [0; 0; 0]^T$ et en passant respectivement par $p_1 = [50; 15; 0]^T$ et $p_2 = [90; 0; 0]^T$. Comme le montre la Figure IV.3, le robot suit parfaitement son chemin prédéfini. L'erreur de suivi de position $E(t)$ est calculée à chaque instant t en utilisant la norme L^2 donnée

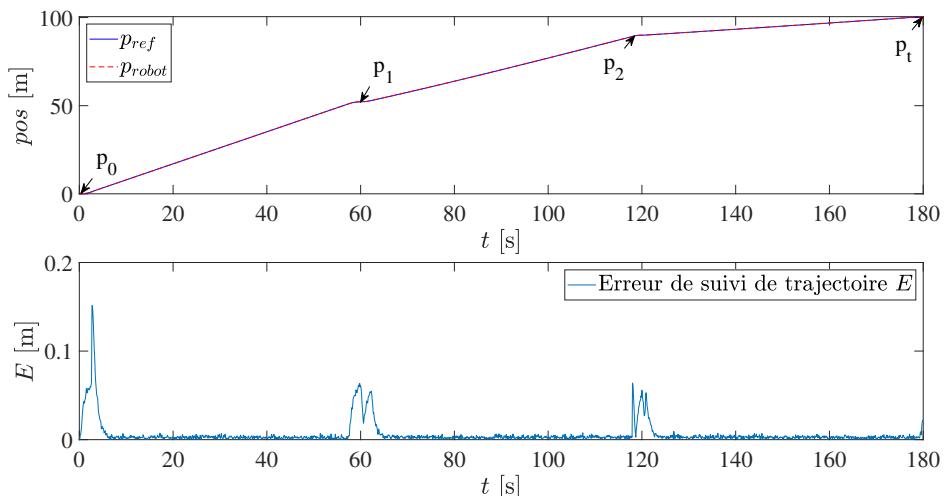


FIGURE IV.3 – Suivi de trajectoire en utilisant la méthode de retour d'état linéarisant.

par l'équation (III.11), Section III.3, avec $T_{robot}(t)$ et $T_{ref}(t)$, représentées ici respectivement par $p_{robot}(t)$ et $p_{ref}(t)$, sont données par (III.12) dans la même Section III.3. L'erreur maximale obtenue

sur le suivi de trajectoire dans cet exemple est de $E_m = 0.1513 \text{ m}$. Il est à noter que les performances du suivi dépendent des coefficients a_i choisis avec la méthode de placement de pôles.

La Figure IV.4 donne le suivi de la position du robot par rapport à l'axe \vec{x} et \vec{y} ($x(t)$ et $y(t)$ respectivement) ainsi que leurs erreurs de suivi correspondantes E_x et E_y . La valeur maximale obtenue sur les erreurs de suivi est la suivante : $\max(|E_x|) = 0,1283 \text{ m}$, et $\max(|E_y|) = 0,0487 \text{ m}$.

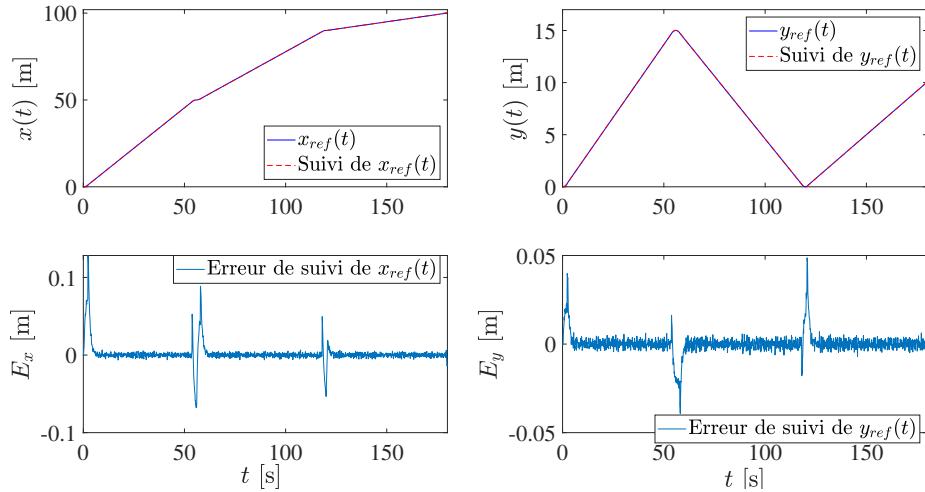


FIGURE IV.4 – Suivi de $x_{ref}(t)$ et de $y_{ref}(t)$ en utilisant la méthode de retour d'état linéarisant.

IV.3.2 Évitement d'obstacles

L'évitement d'obstacles n'est pas l'objectif principal de ce travail. Néanmoins, le but est de pouvoir prendre en considération dans les simulations ce qui se passe réellement dans la fab en cas de rencontre d'un obstacle mobile (*i.e.* les deux comportements adoptés par les robots dans la fab afin d'éviter la collision). Pour cela, nous proposons dans cette partie une stratégie visant à contourner les obstacles imprévus tout en respectant les conditions évolutives de l'environnement des robots.

Afin de définir cette stratégie, les hypothèses suivantes sont faites (voir Figure IV.5 pour l'illustration des variables) :

- 1) Les obstacles sont détectés par des télémètres laser à une distance suffisamment grande, permettant aux robots de décélérer avant d'atteindre la distance de sécurité notée o_1 qui les sépare de l'obstacle.
- 2) Les robots sont équipés de capteurs de navigation (*e.g.* caméra, sonars, ...) fournissant des informations permettant de reconnaître les obstacles et donc d'estimer leurs dimensions o_2 et o_3 .

En se basant sur les hypothèses précédentes et en utilisant la carte de son environnement, le robot est capable de savoir s'il est possible de contourner un obstacle en fonction de l'espace libre qui l'entoure. Si ce n'est pas possible, il s'arrête et attend que l'obstacle s'éloigne de son chemin. À noter que par «obstacles imprévus», nous entendons des obstacles mobiles (tels que des opérateurs

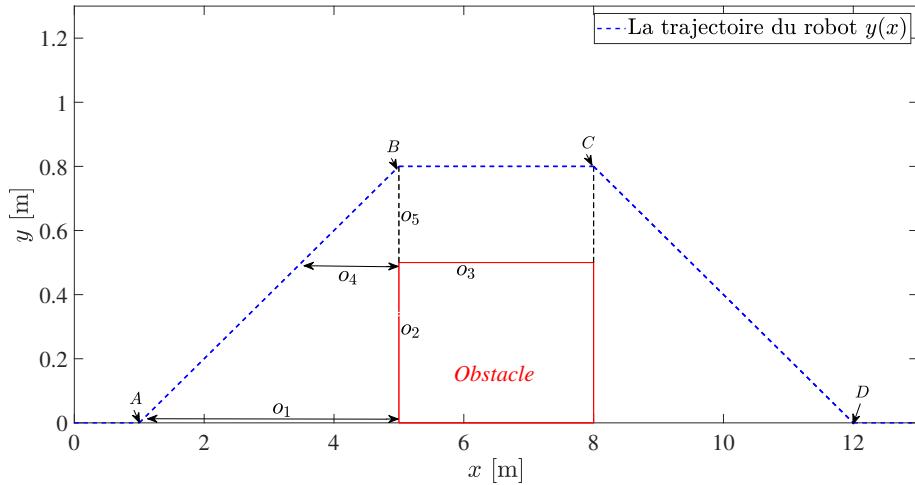


FIGURE IV.5 – Principe de la stratégie d'évitement d'obstacles.

humains, autres robots mobiles, ...) ainsi que des obstacles immobiles non prévus dans le chemin du robot (déplacés par les opérateurs fab tels que des tables ou des chaises,...). À noter que les opérateurs humains de la fab ont reçu des instructions nécessaires pour bien collaborer avec les robots, donc s'ils remarquent qu'un obstacle bloque le robot pour continuer sa tâche (au cas où il ne peut pas le contourner), ils l'éloignent.

Le principe d'évitement d'obstacle adopté dans ce travail est illustré sur la Figure IV.5, où o_4 désigne une distance de sécurité entre le robot et l'obstacle pendant l'opération d'évitement d'obstacle (*i.e.* la distance à laquelle le robot est autorisé à être près de l'obstacle durant le contournement de ce dernier). Cette valeur est fixée par l'utilisateur. Le principe de franchissement d'obstacles est résumé comme suit :

- Après avoir détecté un obstacle, le robot réduit progressivement sa vitesse jusqu'à l'arrêt complet au point A, à une distance notée o_1 de l'obstacle.
- En se basant sur ses estimations, le robot décide s'il est possible de surmonter l'obstacle, et de quel côté.
- **Phase 1** (du point A au point B) : le robot calcule le point B à atteindre en respectant la distance o_4 qui le sépare de l'obstacle durant le contournement, et ceci en utilisant les équations suivantes :

$$\begin{cases} o_5 &= \frac{o_2 o_4}{o_1 - o_4} \\ x_B &= x_A + o_1 \\ y_B &= y_A + o_5 + o_2 \end{cases} \quad (\text{IV.1})$$

- Ensuite, des trajectoires en vitesse et en position sont générées en utilisant l'approche donnée précédemment dans la Section IV.2.2, en respectant la vitesse maximale à ne pas dépasser

lors de l'évitement d'obstacles. Le robot suit ainsi les trajectoires et se déplace en ligne droite jusqu'au point B .

- **Phase 2** (du point B au point C) : lorsque le robot atteint le point B , il continue son déplacement en parcourant la distance de o_3 avec une vitesse constante en suivant une ligne droite.
- **Phase 3** (du point C au point D) : le robot décélère en suivant le profil de trajectoire de vitesse inverse de la phase 1.
- **Phase 4** (à partir du point D) : le robot rejoint son chemin prédéfini (lors de la planification de la mission) et poursuit sa mission.

Il est important de noter que de nombreuses autres méthodes de navigation prenant en compte la problématique de l'évitement d'obstacles avec de meilleures performances peuvent être trouvées dans la littérature (voir les références données dans la Section I.6). La stratégie d'évitement d'obstacles proposée ici peut entraîner un retard plus important par rapport à si une meilleure approche est utilisée. Il est également à noter qu'un retard plus court est plus facile à compenser.

IV.4 Reconfiguration de la trajectoire et gestion du retard

IV.4.1 Le principe de la gestion du retard

Afin de minimiser (ou de complètement compenser) le retard qui peut être généré par les obstacles mobiles durant la navigation, nous proposons une approche basée sur la reconfiguration en temps réel de la trajectoire de vitesse. Cette approche sera détaillée et illustrée dans cette partie.

Comme mentionné précédemment, afin de faciliter la collaboration entre les robots mobiles et les opérateurs de la fab, il est préconisé de ne pas dépasser la valeur recommandée de la vitesse v_{rec} durant le fonctionnement nominal (dans des conditions normales). En cas de rencontre d'un obstacle inattendu sur la trajectoire du robot, et afin d'éviter que le retard généré n'ait un impact significatif sur la mission globale du robot, l'approche proposée consiste à mettre à jour et à reconfigurer les trajectoires de référence, à partir du point de récupération de trajectoire après l'obstacle.

Dans les trajectoires recalculées, le robot est autorisé à atteindre la valeur de la vitesse maximale v_{max} acceptée dans la fab. Cette valeur est supérieure à celle recommandée, et elle est fixée en tenant compte des conditions de travail de la fab, de la sécurité des opérateurs, ainsi que de la limitation des capacités physiques du robot. Les profiles de trajectoire de vitesse et de position sont générés avec le même algorithme donné précédemment dans la Section IV.2.2, en prenant en compte le temps restant avant de finir la mission m_i et la vitesse maximale v_{max} . Une fois le retard compensé, le robot peut réduire sa vitesse et suivre les premières trajectoires planifiées pour le reste de ses missions. L'exemple 1 de la Figure IV.6 illustre ce principe de compensation de retard.

Exemple 1 : Dans cet exemple 1 de la Figure IV.6, le robot doit passer de la position initiale $p_0 = [0; 0; 0]^T$ à la position $p_1 = [50; 15; 0]^T$ en 60 s. Les trajectoires bleues représentent la trajectoire

de référence de vitesse $v_{ref}(t)$ et la trajectoire de position correspondante $p_{ref}(t)$ générées pour le cas des conditions normales de fonctionnement (*i.e.* sans rencontrer d'obstacles inattendus). Ici, et pour cette mission, la vitesse $v_{ref,i}$ est aux alentours de $0,91 \text{ m/s}$. À noter que la vitesse recommandée dans la fab est prise comme étant $v_{rec} = 1,36 \text{ m/s}$, tandis que la vitesse maximale autorisée dans la fab est $v_{max} = 1,8 \text{ m/s}$.

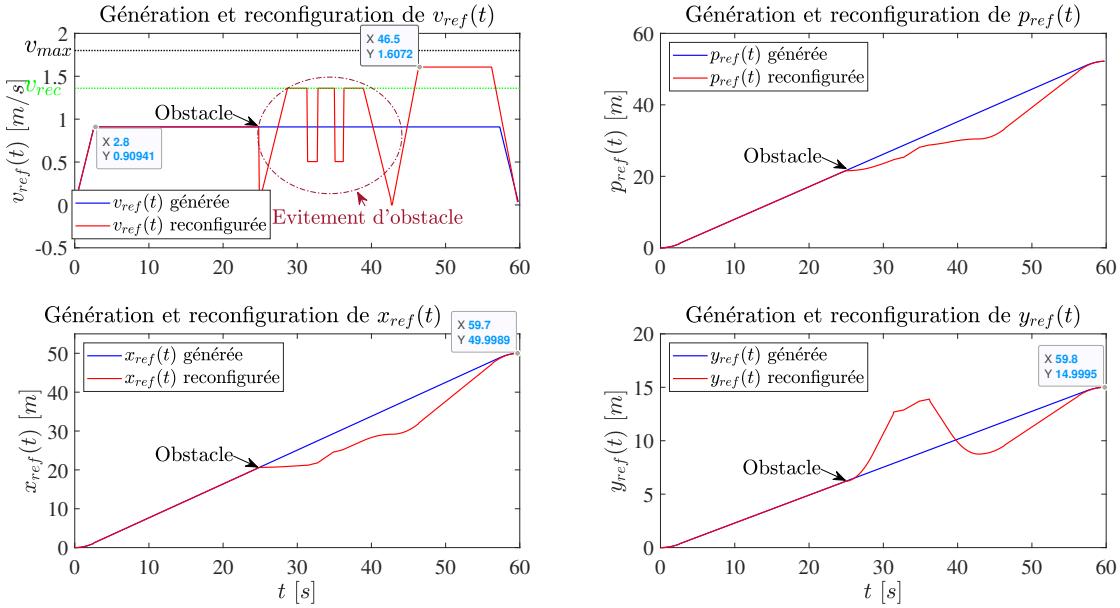


FIGURE IV.6 – Compensation du retard (Exemple 1).

Les trajectoires rouges représentent les trajectoires reconfigurées $v_{ref}(t)$ et $p_{ref}(t)$ après avoir rencontré un obstacle à l'instant 15 s et l'avoir contourné. Comme on peut le voir dans cet exemple, le temps restant avant la fin de la mission après l'évitement d'obstacle est suffisant pour compenser le retard généré par l'obstacle, et ceci avec une vitesse $v_{ref,i} \simeq 1,61, \text{ m/s}$ ($v_{rec} < v_{ref,i} < v_{max}$). Atteindre la valeur v_{max} n'est pas nécessaire mais la valeur v_{rec} n'est pas suffisante pour compenser le retard.

Exemple 2 : Afin de respecter au mieux la durée t planifiée pour la mission globale m , si le retard ne peut pas être complètement compensé au cours de la sous-mission courante m_i , il est transmis à la sous-mission suivante, dans le but de réduire ou de complètement compenser le retard, et ceci en agissant de la même manière sur la trajectoire de vitesse de la sous-mission correspondante. La Figure IV.7 illustre ce principe avec un deuxième exemple (Exemple 2).

Dans cet exemple 2, le robot doit atteindre la position $(p_2 = [90; 0; 0]^T)$ en commençant par la position $(p_0 = [0; 0; 0]^T)$ et en exécutant deux sous-missions :

1. Sous-mission m_1 : Aller de la position p_0 à la position $(p_1 = [50; 15; 0]^T)$ en 60 s .
2. Sous-mission m_2 : Aller de la position p_1 à la position p_2 en 60 autres secondes.

Durant l'exécution de la première sous-mission m_1 , le robot rencontre un obstacle inattendu sur son chemin. Dans cet exemple 2, l'obstacle ne peut pas être évité à cause du manque d'espace. Donc le robot attend 25 s avant que l'obstacle ne s'éloigne de son chemin, puis il continue sa mission.

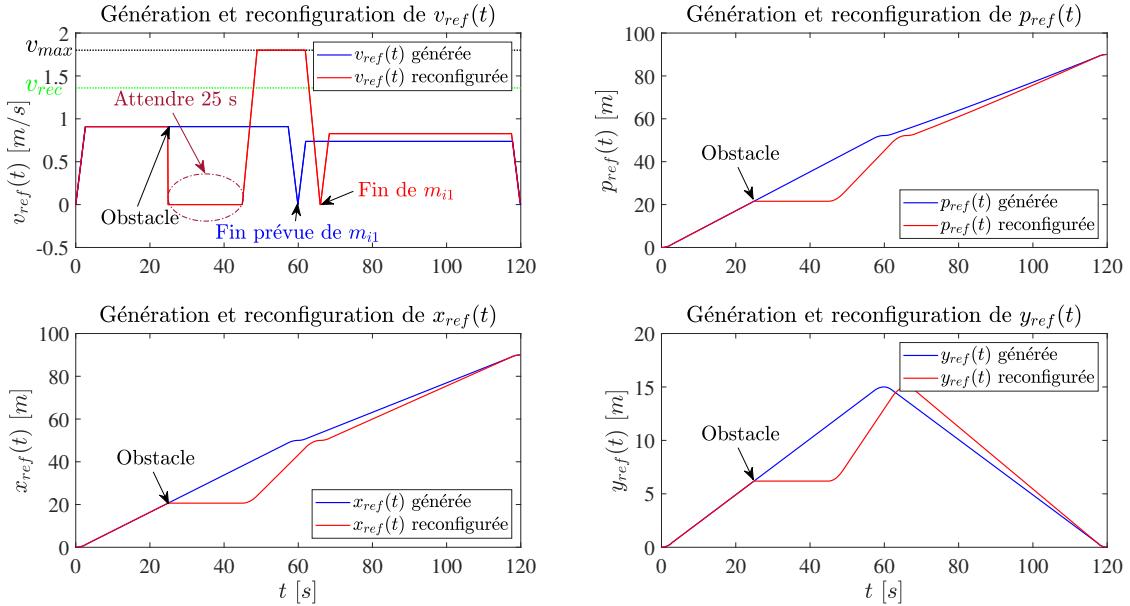


FIGURE IV.7 – Compensation du retard (Exemple 2).

Comme on peut le voir sur la Figure IV.7, v_{max} est atteinte dans la trajectoire de vitesse reconfigurée pour m_1 . Néanmoins, le retard généré par l'obstacle n'a pas pu être compensé lors de cette première sous-mission m_1 . Ainsi, m_1 est accomplie avec un retard. Ce retard est ensuite transmis à la deuxième sous-mission m_2 , et il est pris en compte dans la reconfiguration de la trajectoire de vitesse de m_2 , ceci a permis de compenser le retard. Par conséquent, la mission globale m de cet exemple 2 est réalisée à temps (la durée planifiée pour m a été respectée malgré l'obstacle).

Le principe de l'approche de la gestion du retard est illustré dans le schéma fonctionnel de la Figure IV.8.

IV.4.2 Analyse de la compensation du retard

La compensation du retard dépend de plusieurs paramètres, notamment :

- 1) L'instant d'apparition de l'obstacle.
- 2) Le temps nécessaire pour contourner l'obstacle.
- 3) La distance restante avant d'atteindre la position finale souhaitée.
- 4) Le temps restant par rapport à la durée planifiée avant la fin de la mission.

En tenant compte de ces différents paramètres, et afin de prédire si une mission sera accomplie à temps ou avec un retard, une analyse est effectuée afin d'estimer à chaque instant de navigation le retard maximal qui peut être compensé en cas de rencontre d'un obstacle inattendu. Cela permet

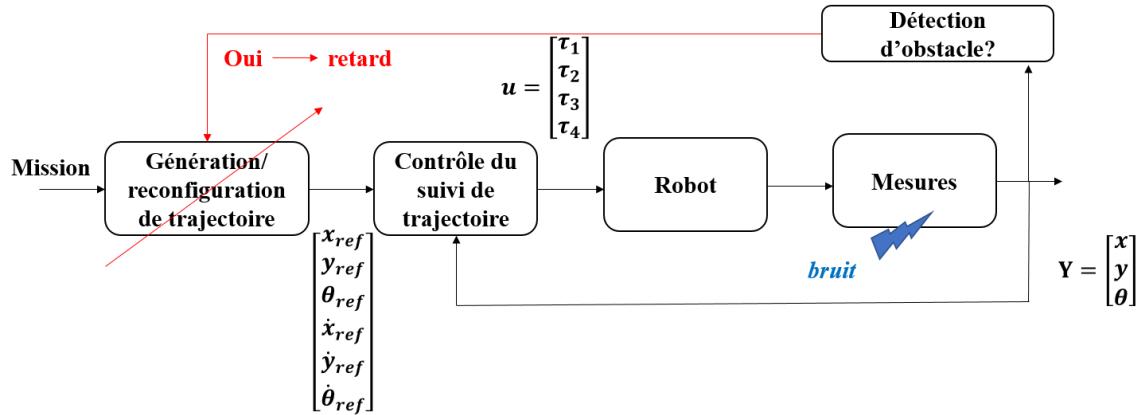


FIGURE IV.8 – Schéma fonctionnel du principe de la compensation du retard.

au gestionnaire de production de mieux surveiller et organiser les missions de transport entre les équipements. Cette analyse est représenté dans la section suivante. A noter qu'avec l'approche de reconfiguration de trajectoire proposée, si un retard ne peut pas être complètement compensé, il est néanmoins réduit et minimisé.

IV.5 Résultats de simulation

Afin de montrer l'efficacité de l'approche de reconfiguration de la trajectoire en vitesse par rapport à la compensation du retard, cette dernière est appliquée au robot omnidirectionnel à 4 roues Mecanum (KUKA-KMP). Nous présentons dans cette partie les résultats de simulation obtenus pour quatre différents scénarios.

Les paramètres physiques du robot sont donnés dans la Table III.4, Section III.4.5. L'accélération et la décélération du robot sont données par : $|acc| = |dec| = 0,35 \text{ m/s}^2$. Quant aux vitesses recommandée et maximale dans la fab, elles sont prises comme étant $v_{rec} = 1,36 \text{ m/s}$ et $v_{max} = 1,8 \text{ m/s}$ comme mentionné dans la Section IV.4. À noter que la vitesse maximale autorisée pendant l'évitement d'obstacles ne doit pas dépasser la vitesse recommandée v_{rec} .

IV.5.1 Les scénarios

Pour illustrer l'efficacité de l'approche proposée, quatre scénarios sont étudiés. Le premier est sans obstacles inattendus et les trois autres sont avec la présence d'obstacles.

1. Scénario #1 : Supposons que le robot doit exécuter une mission globale m , en partant de la position initiale $p_0 = (0; 0; 0)$ vers la position finale souhaitée $p_f = (150; 1; 0)$ en 180 secondes, et ceci en exécutant trois sous-missions selon le scénario suivant :
 - a) Aller de la position p_0 à la position $p_1 = (50; 50; 0)$.
 - b) Aller de la position p_1 à la position $p_2 = (90; 0; 0)$.

c) Aller de la position p_2 à la position p_f .

Chaque sous-mission doit être accomplie en 60 s.

2. Scénario #2 : Le deuxième scénario consiste à exécuter le premier scénario, mais le robot rencontre trois obstacles lors de l'exécution de la mission globale comme suit :
 - a) Un obstacle à l'instant 30 s, contourné par le côté gauche.
 - b) Un obstacle à l'instant 85 s qui ne peut pas être contourné. Le robot attend 10 s jusqu'à ce que l'obstacle s'éloigne de son chemin.
 - c) Un obstacle à l'instant 140 s, contourné par le côté droit.
3. Scénario #3 : Le troisième scénario est similaire au second mais le temps d'attente pour le deuxième obstacle est de 25 s au lieu de 10 s.
4. Scénario #4 : Le dernier scénario a le premier obstacle comme dans le Scénario #2, le deuxième obstacle dure 50 s, et le troisième obstacle n'existe pas.

La première étape consiste à vérifier la faisabilité des sous-missions dans le cas sans obstacles (Scénario #1). Ainsi, pour chaque sous-mission m_i , $t_{min,i}$ est calculé en utilisant l'algorithme donné dans la Section IV.2.2, en tenant compte de la valeur v_{rec} . En effet, $t_{1,min} = 58,2 \text{ s}$, $t_{2,min} = 53 \text{ s}$, et $t_{3,min} = 49,8 \text{ s}$. Toutes les sous-missions ont une durée t_i , ($i = 1,..,3$) sont $t_i > t_{i,min}$. Par conséquent, la mission globale est réalisable.

IV.5.2 Simulations

La Figure IV.9 illustre la trajectoire du robot $y(x)$ (sans et avec les obstacles simulés) avec leurs trajectoires de suivi, en utilisant le contrôle donné précédemment dans la Section III.3. Comme on

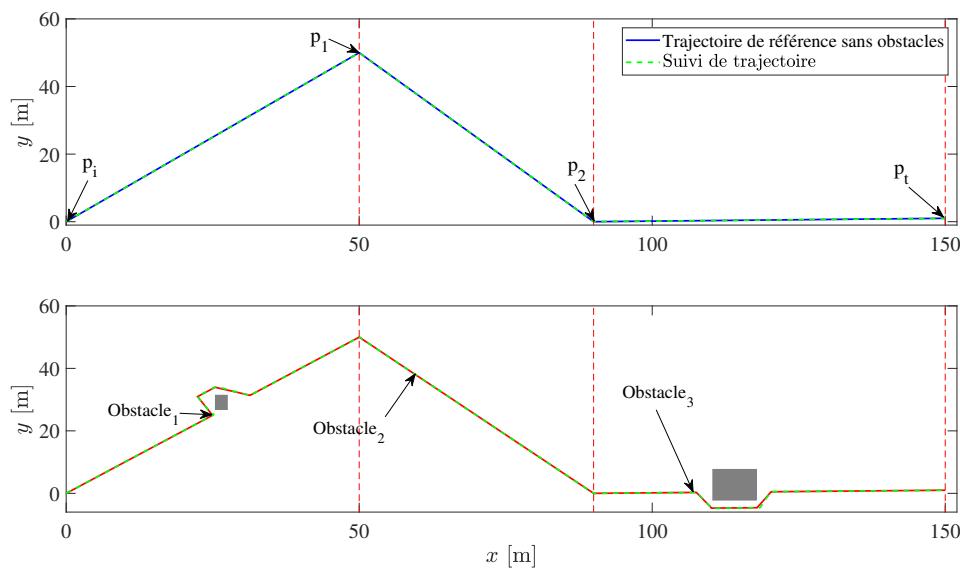


FIGURE IV.9 – Trajectoire de la mission globale sans et avec les obstacles.

peut le voir sur cette Figure IV.9, la loi de commande est efficace même pour l'évitement d'obstacles.

La Figure IV.10 illustre les trajectoires reconfigurées de vitesse $v_{ref}(t)$ en utilisant l'approche proposée, pour les trois derniers scénarios, comparées à celle générée pour le premier scénario, avec les profils déduits pour la position $p_{ref}(t)$.

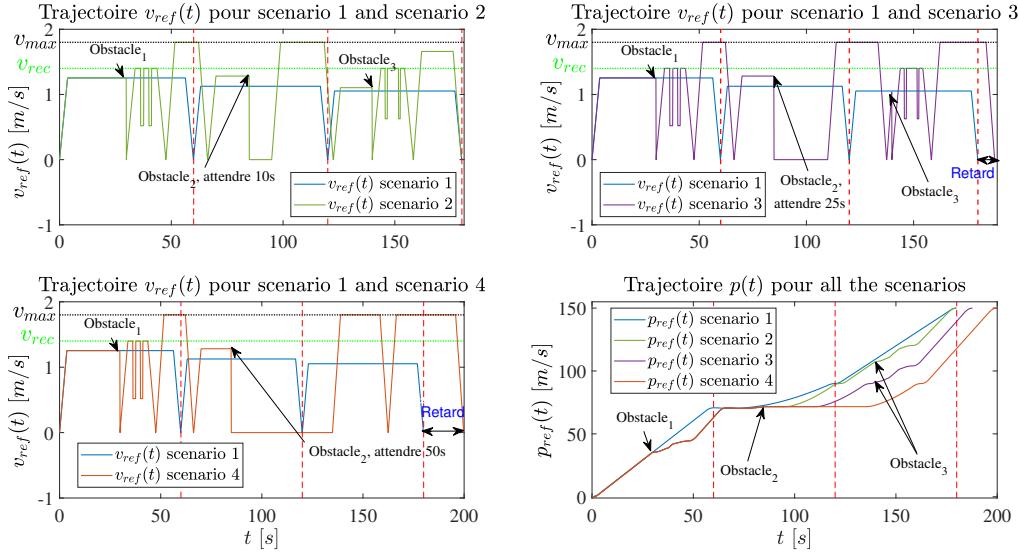


FIGURE IV.10 – Trajectoires de vitesse et de position pour tous les scénarios étudiés.

Pour chaque scénario, les trajectoires $x_{ref}(t)$ et $y_{ref}(t)$ sont obtenues à partir de la projection de la trajectoire par rapport aux axes \vec{x} et \vec{y} , comme illustré dans Figure IV.11.

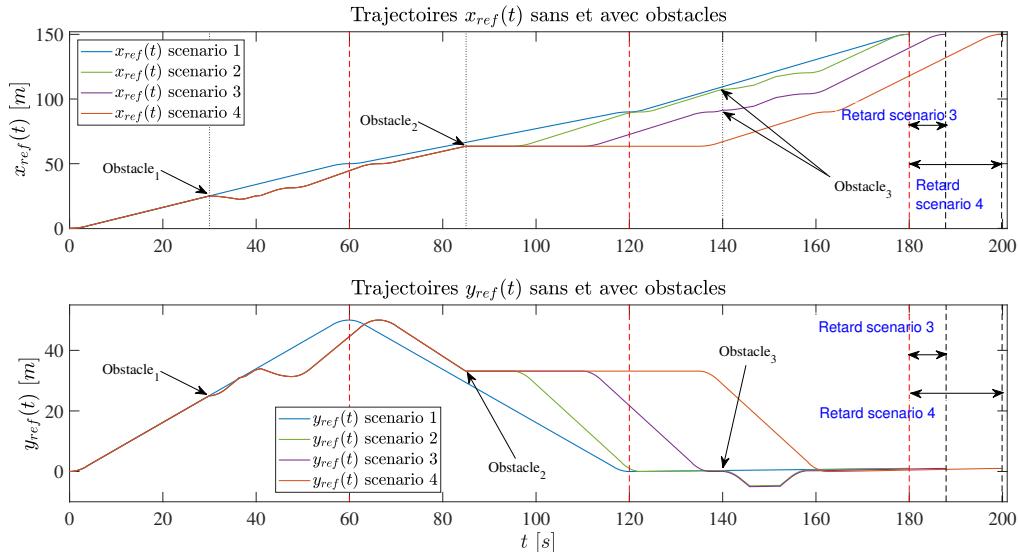


FIGURE IV.11 – Trajectoires $x_{ref}(t)$ et $y_{ref}(t)$ des quatre scénarios étudiés.

Discussion : Comme on peut le constater dans ces résultats, la compensation du retard dépend du temps mis pour contourner l'obstacle. Dans le Scénario #2, le deuxième obstacle ne prend que 10

s avant de s'éloigner du chemin du robot. En utilisant l'approche proposée, le robot est capable de compenser le retard généré par les trois obstacles.

Dans le troisième scénario (Scénario #3), le fait d'attendre 25 s pour que le deuxième obstacle s'éloigne en plus des deux obstacles impacte la durée globale de la mission. La position finale souhaitée est atteinte avec un retard de 7,8 s. Enfin, dans le quatrième scénario, le deuxième obstacle prend 50 s pour s'éloigner. Ce retard ne peut être compensé et la mission se termine avec un retard de 19,8 s même en absence du troisième obstacle.

IV.5.3 Vérification et analyse de la possibilité de compensation du retard

Pour une gestion optimale des missions de transport dans la fab, et afin de prédire si une mission sera accomplie à temps (*i.e.* en respectant sa durée planifiée) ou non, et aussi afin d'estimer à chaque instant le retard maximum qui peut être compensé en cas de rencontre d'un obstacle imprévu sur le chemin du robot, une analyse est effectuée en tenant compte de l'instant d'apparition d'obstacle, du temps et de la distance restants avant d'atteindre la destination finale souhaitée. Cette analyse est présentée dans la Figure IV.12. Un retard nul sur la figure signifie qu'à partir de ce point (la position actuelle), le robot ne peut compenser aucun retard s'il rencontre un obstacle à nouveau. Un retard négatif signifie que le robot arrivera à destination avec un retard, même s'il ne rencontrera plus de nouvel obstacle sur la distance restante à parcourir avant d'atteindre la position désirée. Cette analyse est très importante pour une supervision optimale des missions de transport des FOUPs dans la fab.

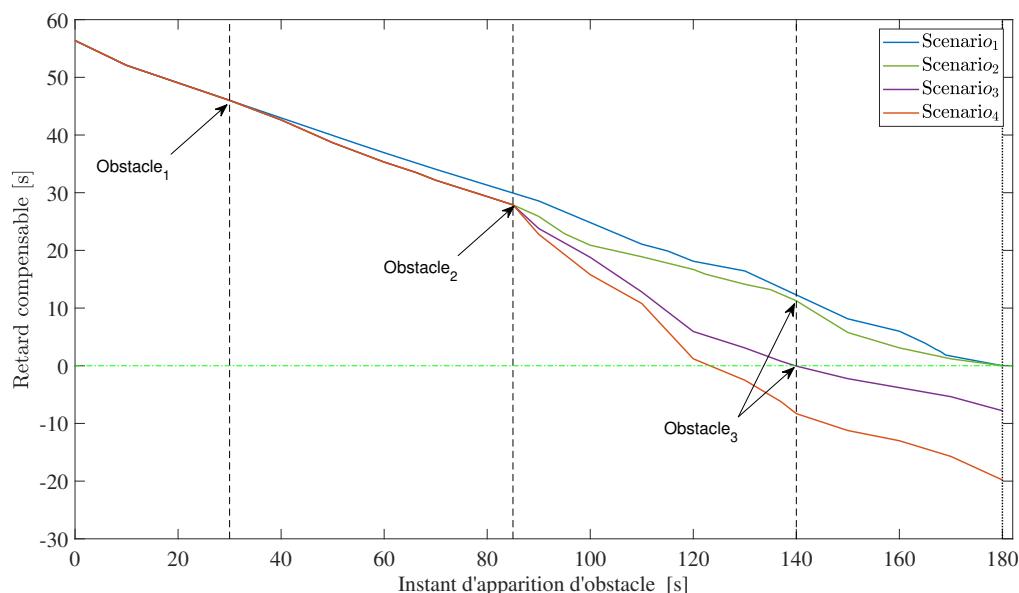


FIGURE IV.12 – Estimation du retard pouvant être compensé à chaque instant pour les quatre scénarios étudiés.

IV.6 Conclusion

Ce chapitre traite la problématique de compensation du retard qui peut être généré par les obstacles mobiles qui apparaissent soudainement sur le chemin des robots mobiles pendant l'exécution de leurs missions dans la fab. Il propose une approche basée sur la mise à jour et la reconfiguration des trajectoires en vitesse et en position des robots en fonction des changements de l'environnement. Sans obstacles, les trajectoires générées respectent la vitesse recommandée. Cette vitesse est fixée par l'utilisateur dans le but de faciliter la collaboration homme / robot dans la fab. Tandis que lorsque le robot rencontre des obstacles inattendus sur son chemin et qui l'empêche de respecter la durée planifiée pour sa mission, le robot est autorisé à atteindre la valeur maximale de la vitesse autorisée dans la fab, en tenant compte du fait que les opérateurs humains de la fab sont formés pour collaborer avec les robots. Cela vise à réduire ou à compenser complètement (quand c'est possible) le retard afin de mieux répondre aux attentes de l'entreprise en matière de robotisation de la fab. A cela, s'ajoute une analyse permettant d'estimer à chaque instant le retard maximal pouvant être compensé en cas d'apparition d'un obstacle. Cela permet de prédire si la mission va être réalisée à temps ou avec un retard, et d'estimer aussi ce retard. Ces informations sont très utiles pour le planificateur des missions de transport et visent à bien encadrer les missions. Ce chapitre s'achève avec des résultats de simulation qui valident l'approche.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Conclusion générale

Le Projet Européen *PRODUCTIVE4.0*, et plus particulièrement le *Work Package WP 8, Section T8.3.4* dénommée *Fab Robotization*, s'intéresse à la robotisation des installations de fabrication des semi-conducteurs dans le but d'améliorer le rendement de production. Il a pour objectifs de contrôler l'état de santé des robots intégrés dans la fab, d'étudier leur performances, et de reconfigurer les robots pour une production optimale. Cette thèse s'inscrit dans le cadre de ce projet, en partenariat avec *STMicroelectronics* de Rousset, un des leaders mondiaux dans la fabrication de semi-conducteurs.

Dans le cadre de ce projet, *STMicroelectronics* intègre des robots mobiles à roues, et plus précisément des robots unicycles et des robots omnidirectionnels à 4 roues Mecanum, afin de transporter des FOUPs entre les équipements de la fab dans le but de :

- Optimiser la production et améliorer le rendement en bénéficiant des avantages que présentent les robots mobiles, notamment la précision, la rapidité, et la capacité de travailler pendant de longues heures sans s'arrêter.
- Moins exposer les opérateurs humains de la fab aux troubles musculosquelettiques qui peuvent être engendrées suite au fait de transporter de lourdes charges d'une façon répétitive.

Malgré les nombreux avantages que présente l'intégration des robots mobiles à roues dans la fab, on distingue deux problématiques majeures qui peuvent empêcher les robots de réaliser leurs missions avec succès ou à temps :

1. Les *défaux*.
2. La rencontre des obstacles mobiles et imprévus, qui ont tendance à générer un retard par rapport au temps d'arrivée des FOUPs au point désiré.

Cette thèse a pour but de répondre à ces deux problématiques en proposant un ensemble de travaux traitants le diagnostic des *défaux* actionneurs et capteurs des deux types de robots d'un côté, et la gestion du retard généré par les obstacles d'un autre côté.

Le premier chapitre de ce manuscrit commence par un état de l'art sur :

- Les différents types des robots à roues.
- Les capteurs utilisés dans la robotique mobile pour la navigation
- Des généralités sur les *défaux* et le diagnostic.

- Les différents travaux proposés dans la littérature afin de diagnostiquer les *défauts* des robots mobiles à roues.
- Des généralités et des références bibliographiques sur la navigation des robots mobiles.

Cet état de l'art nous a permis de positionner nos deux problématiques par rapport à la littérature, et de choisir les méthodes les plus adaptées à nos cas d'étude.

Le deuxième chapitre est consacré aux travaux réalisés dans le but de répondre à la problématique du diagnostic et de l'accommodation des *défauts* actionneurs et capteurs des robots unicycles. L'étape du diagnostic est basée sur l'utilisation d'un banc d'observateurs étendus de Kalman (Extended Kalman Observers ou EKOs) combiné avec de la redondance matérielle sur le capteur de mesure de l'angle d'orientation du robot (gyroscope). La signature des résidus générés avec cette approche est caractéristique de chaque *défaut* étudié. Quant à l'étape de l'accommodation des *défauts*, elle est basée sur l'estimation et la compensation des *défauts*. Les résultats de simulation sont satisfaisants et prometteurs.

Le troisième chapitre décrit les travaux réalisés sur les robots omnidirectionnels à 4 roues Mecanum (4-MWMRs). Le premier travail traite le problème du diagnostic des *défauts* actionneurs et capteurs à la fois. L'approche proposée est basée sur l'utilisation d'un banc d'observateurs étendus de Kalman (EKOs) et la généraion d'un ensemble de résidus. Avec cette méthode, l'isolation des *défauts* capteurs est facile et elle est basée sur la signature des résidus. Quant aux *défauts* actionneurs, plus de caractéristiques sur les résidus (*i.e.* la rapidité de détection et le pourcentage de détection) sont pris en compte pour l'isolation.

Un deuxième travail visant à isoler les *défauts* actionneurs plus facilement est proposé par la suite. Il est basé sur l'utilisation d'un banc d'observateurs à entrées inconnues (Unknown Input Oservers ou UIOs) pour les systèmes linéaires à paramètres variants (Linear Parameter Variant Systems ou LPV Systems), et sur la génération d'un ensemble de résidus avec une signature caractéristique de chaque *défaut*.

Nous nous intéressons par la suite dans le même chapitre à la dégradation de tous les actionneurs des robots 4-MWMRs à la fois, suite à la sollicitation importante des robots dans la fab, et nous proposons une approche permettant d'étudier l'impact de ces dégradations sur le comportement des robots. En fonction de l'impact de ces dégradations sur le comportement des robots, des limites de sécurité sont déterminées afin de définir quel niveau de dégradation peut être atteint sans que cela impacte le comportement des robots. Cela a pour but de fournir à l'utilisateur une aide à la décision (Decision-making support) afin d'éviter d'arrêter en urgence les robots après une première détection de *défauts* au niveau des actionneurs quand ce *défaut* n'a pas un impact considérable sur le comportement des robots.

Le quatrième chapitre est dédié à la présentation des travaux proposés afin de répondre à la problématique de gestion du retard généré par les obstacles mobiles et imprévus sur le chemin des robots. Une approche basée sur la reconfiguration de trajectoire en temps réel est proposée. Cette reconfiguration se fait en fonction du retard généré et en respectant la vitesse maximale à ne pas

dépasser dans la fab. Afin de prévoir si une mission va être accomplie à temps ou pas, une analyse est faite afin d'estimer le retard qui peut être compensé à chaque instant de navigation, en fonction de l'instant d'apparition de l'obstacle, de la distance et du temps restant avant d'atteindre la position finale, ainsi que du temps nécessaire pour contourner l'obstacle. Dans le cas où le retard ne peut pas être complètement compensé, cette analyse permet d'estimer le retard à la fin de la mission. Elle permet donc de bien superviser les missions dans la fab.

Perspectives

Dans ce mémoire, nous avons mis en place des approches de diagnostic à base de modèle afin de détecter et d'isoler les *défauts* capteurs et actionneurs sur les deux types de robots considérés (robots unicycles et robots omnidirectionnels à 4 roues Mecanum), ainsi qu'une méthodologie permettant de superviser l'état de santé des actionneurs et d'étudier son impact sur le comportement du robot d'un côté, et une approche permettant de compenser ou de minimiser le retard généré par les obstacles afin d'optimiser la production d'un autre côté.

L'objectif premier était de faire une démonstration quant à la faisabilité, autrement une preuve de concept (Proof Of Concept ou POC) de ces approches et méthodologies. Pour cela, nous avons extrait de la littérature les modèles mathématiques des deux types de robots, et nous avons simulé nos approches afin de montrer leur capacité de répondre aux problématiques considérées. Comme perspectives, nous nous intéressons à tester et à valider ces approches sur les robots intégrés dans la fab.

Pour cela, il faut dans un premier temps valider les modèles mathématiques des deux robots. En effet, la structure des modèles considérés a été validée dans la littérature sur des robots du même type [11, 148, 196], tandis que les paramètres utilisés ne sont pas tous fournis par le constructeur. Par exemple, les moments d'inertie ont été estimés en se basant sur les paramètres connus (masse du robot, ...). Le but est d'identifier les paramètres manquants et de quantifier l'erreur associée au modèle. Cela nous permettra ensuite de tester et de valider nos approches de diagnostic.

Dans un second temps, afin d'exploiter la méthodologie visant à définir des limites de dégradation au niveau des actionneurs, nous devons avoir la limite autorisée de la déviation maximale du robot par rapport à la trajectoire de référence S_l auprès d'un expert de la fab. En effet, dans un environnement tel que la fab, il peut y avoir des zones plus confinées que d'autres. Dans ce cas, on peut avoir une limite qui varie selon la zone de fonctionnement des robots. Avoir les bonnes valeurs de S_l nous permettra d'adapter notre méthodologie sur l'environnement réel et d'exploiter les résultats. Cette dernière pourra également être étendue au niveau flotte afin de gérer les robots avec des restrictions respectant l'état de santé de chaque robot.

Dans le cadre de cette thèse, nous nous sommes plus focalisés sur les robots omnidirectionnels. Dans un troisième temps, nous pouvons adapter les mêmes méthodologies de surveillance de l'état de santé des actionneurs et de gestion du retard aux robots unicycles.

Enfin, nous pouvons intégrer l'approche de gestion du retard au niveau flotte afin de réaffecter les missions des robots d'une façon optimale de telle sorte à avoir le minimum du retard sur le transport des produits.

BIBLIOGRAPHIE

- [1] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. Trajectory reconfiguration for time delay reduction in the case of unexpected obstacles : application to 4-mecanum wheeled mobile robots (4-MWMR) for industrial purposes. In *IFAC World Congress 2020*, Berlin, Germany, July 2020. URL <https://hal-amu.archives-ouvertes.fr/hal-02799564>.
- [2] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. 4-mecanum wheeled mobile robot actuator fault detection & isolation using unknown input observer-based approach. In *European Control Conference (ECC)*, pages 148–153. IEEE, 2020. doi : <https://hal.archives-ouvertes.fr/hal-02454947v2>.
- [3] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. Trajectory tracking and time delay management of 4-mecanum wheeled mobile robots (4-MWMR). In *ACD'19 : 15th European Workshop on Advanced Control and Diagnosis*, Bologne, Italy, November 2019. URL <https://hal.archives-ouvertes.fr/hal-02381870>.
- [4] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. Mobile robot additive fault diagnosis and accommodation. In *8th International Conference on Systems and Control (ICSC)*, pages 241–246. IEEE, 2019.
- [5] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. Detection & isolation of sensor and actuator additive faults in a 4-mecanum wheeled mobile robot (4-mwmr). In *International Conference on Control, Automation and Diagnosis (ICCAD)*, 2019.
- [6] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. On fault detection and isolation applied on unicycle mobile robot sensors and actuators. In *7th International Conference on Systems and Control (ICSC)*, pages 148–153. IEEE, 2018.
- [7] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. Actuator health state monitoring & degradation impact study on a 4-mecanum wheeled mobile robot behaviour. In *American Control Conference ACC*, New Orleans, Louisiana, USA, May 2021.
- [8] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. Health state monitoring of 4-mecanum wheeled mobile robot actuators and its impact on the robot behaviour analysis. In *Journal of Intelligent & Robotic Systems*, .
- [9] S. Mellah, G. Graton, M. El Adel, M. Ouladsine, and A. Planchais. Trajectory reconfiguration for unexpected obstacles generated time delay reduction : application to 4-mecanum wheeled mobile robots for industrial purposes. In *AIEEE Transactions on Intelligent Transportation Systems*, .
- [10] Téwendé Kiba. Simulation et optimisation du transport automatisé dans la fabrication de semi-conducteurs. *Thèse de doctorat. Ecole Nationale Supérieure des Mines de Saint-Etienne*, 2010.
- [11] P. Vlantis, C. P. Bechlioulis, G. Karras, G. Fourlas, and K. J. Kyriakopoulos. Fault tolerant control for omni-directional mobile platforms with 4 mecanum wheels. In *IEEE ICRA*, pages 2395–2400, May 2016.
- [12] Joaquim A Batlle and Ana Barjau. Holonomy in mobile robots. *Robotics and Autonomous Systems*, 57(4) :433–440, 2009.
- [13] G. Campion, G. Bastin, and B. Dandrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1) :47–62, 1996.

- [14] Bengt Erland Ilon. Wheels for a course stable selfpropelling vehicle movable in any desired direction on the ground or some other base, April 8 1975. US Patent 3,876,255.
- [15] F. G. Pin and S. M. Kilough. A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *IEEE Transactions on Robotics and Automation*, 10(4) :480–489, 1994.
- [16] M. West and H. Asada. Design and control of ball wheel omnidirectional vehicles. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 1931–1938 vol.2, 1995.
- [17] S. Ostrovskaya, J. Angeles, and R. Spiteri. Dynamics of a mobile robot with three ball-wheels. *The International Journal of Robotics Research*, 19(4) :383–393, 2000. doi : 10.1177/02783640022066923. URL <https://doi.org/10.1177/02783640022066923>.
- [18] Olaf Diegel, Aparna Badve, Glen Bright, Johan Potgieter, and Sylvester Tlale. Improved mecanum wheel design for omni-directional robots. 2002.
- [19] G. Indiveri. Swedish wheeled omnidirectional mobile robots : Kinematics analysis and control. *IEEE Transactions on Robotics*, 25(1) :164–171, 2009.
- [20] Jun Qian, Bin Zi, Daoming Wang, Yangang Ma, and Dan Zhang. The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system. *Sensors*, 17(9) :2073, 2017.
- [21] Kiattisin Kanjanawanishkul. Omnidirectional wheeled mobile robots : wheel types and practical applications. *International Journal of Advanced Mechatronic Systems*, 6(6) :289–302, 2015.
- [22] G. Campion, G. Bastin, and B. Dandrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1) :47–62, 1996.
- [23] Rached Dhaouadi and A Abu Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies : A unified framework. *Advances in Robotics & Automation*, 2(2) :1–7, 2013.
- [24] Jakub Cerkala and Anna Jadlovská. Nonholonomic mobile robot with differential chassis mathematical modelling and implementation in simulink with friction in dynamics. In *Acta Electrotechnica et Informatica*, volume 5, pages 3–8, 2015.
- [25] Jae-Bok Song and Kyung-Seok Byun. Steering control algorithm for efficient drive of a mobile robot with steerable omni-directional wheels. *Journal of mechanical science and technology*, 23 (10) :2747, 2009.
- [26] Masayoshi Wada and Shunji Mori. Holonomic and omnidirectional vehicle with conventional tires. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3671–3676. IEEE, 1996.
- [27] D. Filliat. Robotique mobile. *Cours C10-2, ENSTA. Paris, France*, 2005.
- [28] H. R. Everett. *Sensors for Mobile Robots*. A K PETERS. ISBN 1568810482, 9781568810485. URL <http://gen.lib.rus.ec/book/index.php?md5=8a552b7b27fe98883de27ebddafe478e>.
- [29] J. Borenstein, H.R. Everett, and L. Feng. Where am i? sensors and methods for mobile robot positioning. *University of Michigan*, 119(120) :27, 1996.
- [30] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe. Mobile robot positioning : Sensors and techniques. *Journal of robotic systems*, 14(4) :231–249, 1997.
- [31] Anita M Flynn. Combining sonar and infrared sensors for mobile robot navigation. *The International Journal of Robotics Research*, 7(6) :5–14, 1988.
- [32] M. Kam, Xiaoxun Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*, 85(1) :108–119, 1997.
- [33] Bernard Bayle. Robotique mobile. *Cours, Université de Strasbourg, Tech. Rep*, 2008.

- [34] Kok Seng Chong and L. Kleeman. Accurate odometry and error modelling for a mobile robot. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 2783–2788 vol.4, 1997.
- [35] T. Bailey, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte. Data association for mobile robot navigation : a graph theoretic approach. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 3, pages 2512–2517 vol.3, 2000.
- [36] J. A. Perez, J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardos. Continuous mobile robot localization : vision vs. laser. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4, pages 2917–2923 vol.4, 1999.
- [37] L. Kleeman. Advanced sonar and odometry error modeling for simultaneous localisation and map building. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 1, pages 699–704 vol.1, 2003.
- [38] A. J. Davison and N. Kita. 3d simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.
- [39] D. Santosh, S. Achar, and C. V. Jawahar. Autonomous image-based exploration for mobile robot navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 2717–2722, 2008.
- [40] K. Okuyama, T. Kawasaki, and V. Kroumov. Localization and position correction for mobile robot using artificial visual landmarks. In *The 2011 International Conference on Advanced Mechatronic Systems*, pages 414–418, 2011.
- [41] Y. Watanabe and S. Yuta. Position estimation of mobile robots with internal and external sensors using uncertainty evolution technique. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2011–2016 vol.3, 1990.
- [42] C. B. Low and D. Wang. Gps-based path following control for a car-like wheeled mobile robot with skidding and slipping. *IEEE Transactions on Control Systems Technology*, 16(2) :340–347, 2008.
- [43] I. Ashokaraj, A. Tsourdos, P. Silson, and B. White. Sensor based robot localisation and navigation : using interval analysis and extended kalman filter. In *2004 5th Asian Control Conference (IEEE Cat. No.04EX904)*, volume 2, pages 1086–1093 Vol.2, 2004.
- [44] Leopoldo Jetto, Sauro Longhi, and Giuseppe Venturini. Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation*, 15(2) :219–229, 1999.
- [45] R. Isermann and P. Ballé. Trends in the application of model based fault detection and diagnosis of technical processes. In *IFAC Proceedings Volumes*, volume 29, pages 6325–6336, 06 1996.
- [46] Fredrik Gustafsson and Fredrik Gustafsson. *Adaptive filtering and change detection*, volume 1. Citeseer, 2000.
- [47] Ehsan Sobhani-Tehrani and Khashayar Khorasani. *Fault diagnosis of nonlinear systems using a hybrid approach*, volume 383. Springer Science & Business Media, 2009.
- [48] Rolf Isermann. Supervision, fault-detection and fault-diagnosis methods—an introduction. *Control engineering practice*, 5(5) :639–652, 1997.
- [49] Moamar Sayed-Mouchaweh and Patrice Billaudel. Abrupt and drift-like fault diagnosis of concurrent discrete event systems. In *2012 11th International Conference on Machine Learning and Applications*, volume 2, pages 434–439. IEEE, 2012.
- [50] Methnani Salowa. Diagnostic, reconstruction et identification des défauts capteurs et actionneurs : application aux station d'épurations des eaux usées. *Thèse de doctorat de l'université de Toulon et du Var; Ecole nationale d'ingénieurs de Sfax*, 2012.

- [51] Paul M Frank and Birgit Köppen-Seliger. New developments using ai in fault diagnosis. *Engineering Applications of Artificial Intelligence*, 10(1) :3–14, 1997.
- [52] Rolf Isermann. *Fault-diagnosis systems : an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.
- [53] MJ Chantier, George Macleod Coghill, Qiang Shen, and RR Leitch. Selecting tools and techniques for model-based diagnosis. *Artificial Intelligence in Engineering*, 12(1-2) :81–98, 1998.
- [54] Sourabh Dash and Venkat Venkatasubramanian. Challenges in the industrial applications of fault diagnostic systems. *Computers & chemical engineering*, 24(2-7) :785–791, 2000.
- [55] R. Rengaswamy V. Venkatasubramanian, Y. Kewen, and N. S Kavuri. A review of process fault detection and diagnosis : Part i : Quantitative model-based methods. *Computers & Chemical Engineering*, 2003.
- [56] Rolf Isermann. *Fault-Diagnosis Applications : Model-Based Condition Monitoring Actuators, Drives, Machinery, Plants, Sensors, and Fault-Tolerant Systems*. Springer Publishing Company, Incorporated, 2014. ISBN 3642434762.
- [57] Alexandros Mouzakitis. Classification of fault diagnosis methods for control systems. *Measurement and Control*, 46(10) :303–308, 2013.
- [58] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis : Part iii : Process history based methods. *Computers & Chemical Engineering*, 2003.
- [59] V. Venkatasubramanian, R. Rengaswamy, and N. S Kavuri. A review of process fault detection and diagnosis part ii : Qualitative models and search strategies. 03 2003.
- [60] FA Evans and JAMES C WILCOX. Experimental strapdown redundant sensor inertial navigation system. *Journal of Spacecraft and Rockets*, 7(9) :1070–1074, 1970.
- [61] Kevan C. Daly, Eliezer G. Gai, and James V. Harrison. Generalized likelihood test for fdi in redundant sensor configurations. 1979.
- [62] L Mironovskii. Functional diagnosis of dynamic systems(survey). *AUTOM. & REMOTE CONTR.*, 41(8) :1122–1143, 1980.
- [63] E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7) :603–614, 1984.
- [64] Rolf Isermann. Process fault diagnosis based on modeling and estimation methods -a survey. *Automatica*, 20 :387–404, 1984.
- [65] R. F. Constantinescu, P. D. Lawrence, P. G. Hill, and T. S. Brown. Model-based fault diagnosis of a two-stroke diesel engine. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, volume 3, pages 2216–2220 vol.3, 1995.
- [66] L. Dinca, T. Aldemir, and G. Rizzoni. A model-based probabilistic approach for fault detection and identification with application to the diagnosis of automotive engines. *IEEE Transactions on Automatic Control*, 44(11) :2200–2205, 1999.
- [67] Tao Jiang, Khashayar Khorasani, and Siamak Tafazoli. Parameter estimation-based fault detection, isolation and recovery for nonlinear satellite models. *IEEE Transactions on control systems technology*, 16(4) :799–808, 2008.
- [68] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.
- [69] D. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6) :596–602, 1971.
- [70] Clement Treangle. Observateurs grand gain pour des systèmes non linéaires à sorties échantillonnées et retardées. *Thèse de doctorat en Automatique / Robotique*. Normandie Université, 2018.

- [71] Arthur J Krener, Alberto Isidori, et al. Linearization by output injection and nonlinear observers. *Systems & Control Letters*, 3(1) :47–52, 1983.
- [72] Arthur J Krener and Witold Respondek. Nonlinear observers with linearizable error dynamics. *SIAM Journal on Control and Optimization*, 23(2) :197–216, 1985.
- [73] Xiao-Hua Xia and Wei-Bin Gao. Nonlinear observer design by observer error linearization. *SIAM Journal on Control and Optimization*, 27(1) :199–216, 1989.
- [74] M Hou and AC Pugh. Observer with linear error dynamics for nonlinear multi-output systems. *Systems & Control Letters*, 37(1) :1–9, 1999.
- [75] M. Guay. Observer linearization by output-dependent time-scale transformations. *IEEE Transactions on Automatic Control*, 47(10) :1730–1735, 2002.
- [76] Driss Boutat, Abderraouf Benali, Hassan Hammouri, and Krishna Busawon. New algorithm for observer error linearization with a diffeomorphism on the outputs. *Automatica*, 45(10) :2187–2193, 2009.
- [77] R. Rajamani. Observers for lipschitz nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3) :397–401, 1998.
- [78] Frédéric Mazenc, Silviu-Iulian Niculescu, and Olivier Bernard. Exponentially stable interval observers for linear systems with delay. *SIAM Journal on Control and Optimization*, 50(1) :286–305, 2012.
- [79] Denis Efimov and Tarek Raïssi. Design of interval observers for uncertain dynamical systems. *Automation and Remote Control*, 77(2) :191–225, 2016.
- [80] Michael Zeitz. The extended luenberger observer for nonlinear systems. *Systems & Control Letters*, 9(2) :149–156, 1987.
- [81] Fatiha Nejjari, Vicenc Puig, L Giancristofaro, and S Koehler. Extended luenberger observer-based fault detection for an activated sludge process. *IFAC Proceedings Volumes*, 41(2) :9725–9730, 2008.
- [82] Chuei-Tin Chang and Jen-Wen Chen. Implementation issues concerning the ekf-based fault diagnosis techniques. *Chemical Engineering Science*, 50(18) :2861–2882, 1995.
- [83] Nicolas Boizot, Eric Busvelle, and Jean-Paul Gauthier. An adaptive high-gain observer for nonlinear systems. *Automatica*, 46(9) :1483–1488, 2010.
- [84] Jie Chen, Ron J Patton, and Hong-Yue Zhang. Design of unknown input observers and robust fault detection filters. *International Journal of control*, 63(1) :85–105, 1996.
- [85] Elena Dubrova. *Hardware Redundancy*, pages 55–86. Springer New York, New York, NY, 2013. ISBN 978-1-4614-2113-9. doi : 10.1007/978-1-4614-2113-9_4. URL https://doi.org/10.1007/978-1-4614-2113-9_4.
- [86] J. Gertler. Fault detection and diagnosis. *Encyclopedia of Systems and Control*, 2014. doi : DOI10.1007/978-1-4471-5102-9_223-1.
- [87] F. Baghernezhad and K. Khorasani. Computationally intelligent strategies for robust fault detection, isolation, and identification of mobile robots. *Neurocomput.*, 171(C) :335–346, 2016. doi : 10.1016/j.neucom.2015.06.050.
- [88] J. Chen and R. J. Patton. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, 1999.
- [89] F. Pukelsheim. The three sigma rule. *The American Statistician*, 48(2) :88–91, 1994.
- [90] P. M. Frank. Fault diagnosis in dynamic systems via state estimation-a survey. In *System fault diagnostics, reliability and related knowledge-based approaches*, pages 35–98. Springer, 1987.
- [91] R.N. Clark. Instrument fault detection. In *IEEE Trans. Aerospace and Electronic Systems*, pages 456–465. Springer, 1978.

- [92] D. Zhuohua, C. Zixing, and Y. Jinxia. Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments : A survey. In *Proceedings of IEEE ICRA, International Conference on Robotics and Automation*, 2005.
- [93] M. Luo, D. Wang, M. Pham, J.B. Zhang, D.H. Zhang, and Y.Z. Zhao. Model-based fault diagnosis/prognosis for wheeled mobile robots : a review. *31st Annual Conference of IEEE Industrial Electronics Society, IECON*, 2005.
- [94] K. Anastassia and P. Plöger. Model-based fault diagnosis techniques for mobile robots. *IFAC-PapersOnLine*, 49 :50–56, 2016.
- [95] JOQUÍN L. FERNÁNDEZ, RAFAEL SANZ, and AMADOR R. DIÉGUEZ. Probabilistic models for monitoring and fault diagnosis : application and evaluation in a mobile robot. *Applied Artificial Intelligence*, 18(1) :43–67, 2004. doi : 10.1080/08839510490250097.
- [96] J. Carlson and R. R. Murphy. An investigation of mml methods for fault diagnosis in mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.
- [97] E. N. Skoundrianos and S. G. Tzafestas. Finding fault - fault diagnosis on the wheels of a mobile robot using local model neural networks. *IEEE Robotics & Automation Magazine*, 2004.
- [98] Michèle Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes : theory and application*, volume 104. prentice Hall Englewood Cliffs, 1993.
- [99] G. Steinbauer M. Brandstötter, M. W. Hofbaur and F. Wotawa. Model-based fault diagnosis and reconfiguration of robot drives. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- [100] Gerald Steinbauer, Martin Mörtl, and Franz Wotawa. Real-time diagnosis and repair of faults of robot control software. In Ansgar Bredenfeld, Adam Jacoff, Itsuki Noda, and Yasutake Takahashi, editors, *RoboCup 2005 : Robot Soccer World Cup IX*, pages 13–23. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-35438-3.
- [101] Zheng Li. Fault diagnosis and fault tolerant control of mobile robot based on neural networks. *International Conference on Machine Learning and Cybernetics*, 2009.
- [102] N. Hoang and H. Kang. A model-based fault diagnosis scheme for wheeled mobile robots. *Int. J. Control Autom. Syst.*, 2014. doi : <https://doi.org/10.1007/s12555-013-0012-1>.
- [103] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. Sensor fault detection and identification in a mobile robot. In *Proceedings. IEEE/RSJ ICIRS . Innovations in Theory, Practice and Applications*, Oct 1998.
- [104] S. I. Roumeliotis P. Goel, G. Dedeoglu and G. S. Sukhatme. Fault detection and identification in a mobile robot using multiple model estimation and neural network. *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.
- [105] L. Yutian and C. Jungan. Integrated fault diagnosis method of mobile robot. *Communications in Computer and Information Science.*, 164 :372–379, 01 2011.
- [106] G.K. Fourlas, G.C. Karras, and K.J. Kyriakopoulos. Sensors fault diagnosis in autonomous mobile robots using observer based technique. In *ICRA*, May 2015.
- [107] C. Zixing, D. Zhuohua, C. Jingfeng, Z. Xiaobing, and Y. Jinxia. A multiple particle filters method for fault diagnosis of mobile robot dead-reckoning system. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [108] T. Nakagami M. Hashimoto, H. Kawashima and F. Oba. Sensor fault detection and identification in dead-reckoning system of mobile robot : interacting multiple model approach. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.
- [109] H. Kawashima M. Hashimoto and F. Oba. A multi-model based fault detection and diagnosis of internal sensors for mobile robot. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

- [110] Michał Zająć, Dariusz Uciński, and Ralf Stetter. Mobile robot diagnosis with bayesian filters. In *Mechatronic Systems and Materials III*, volume 147 of *Solid State Phenomena*, pages 518–523. Trans Tech Publications Ltd, 4 2009. doi : 10.4028/www.scientific.net/SSP147-149.518.
- [111] J.L. LIN and J.P. JIANG. Diagnosis of simultaneous faults for mobile robots based on fuzzy clustering method. *International Conference on Machine Learning and Cybernetics*, 2010.
- [112] G.K. Fourlas, S. Karkanis, G.C. Karras, and K.J. Kyriakopoulos. Model based actuator fault diagnosis for a mobile robot. In *IEEE International Conference on Industrial Technology (ICIT)*, Feb 2014.
- [113] D. Stavrou, D.G. Eliades, C.G. Panayiotou, and M.M. Polycarpou. Fault detection for service mobile robots using model-based method. *Autonomous Robots SP*, pages 1–20, 2016. doi : <https://doi.org/10.1007/s10514-015-9475-7>.
- [114] P. Yazdjerdi and N. Meskin. Actuator fault detection and isolation of differential drive mobile robots using multiple model algorithm. In *4th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 0439–0443. IEEE, 2017.
- [115] Zsolt Kira. Modeling cross-sensory and sensorimotor correlations to detect and localize faults in mobile robots. *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- [116] B. Abci, M.B. El Najjar, V.G. Cocquempot, and Dherbomez. An informational approach for sensor and actuator fault diagnosis for autonomous mobile robots. *Journal of Intelligent & Robotic Systems*, pages 1–20, 2019.
- [117] T.A. Baede. Motion control of an omnidirectional mobile robot. In *Traineeship Report DCT, National University of Singapore, Faculty of Engineering, Department of Mechanical Engineering*, 2006.
- [118] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [119] John Canny. *The complexity of robot motion planning*. MIT press, 1988.
- [120] Kikuo Fujimura. *Motion planning in dynamic environments*. Springer Science & Business Media, 2012.
- [121] Yong K Hwang and Narendra Ahuja. Gross motion planning—a survey. *ACM Computing Surveys (CSUR)*, 24(3) :219–291, 1992.
- [122] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.
- [123] Jacob T Schwartz and Chee-Keng Yap. *Advances in robotics : Algorithmic and geometric aspects of robotics*. volume 1. 1986.
- [124] Jianjun Ni, Liuying Wu, Xinnan Fan, and Simon X Yang. Bioinspired intelligent algorithm and its applications for mobile robot control : a survey. *Computational intelligence and neuroscience*, 2016.
- [125] Keiji Nagatani, Howie Choset, and Sebastian Thrun. Towards exact localization without explicit localization with the generalized voronoi graph. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 1, pages 342–348. IEEE, 1998.
- [126] Lozano-Perez. Spatial planning : A configuration space approach. *IEEE Transactions on Computers*, C-32(2) :108–120, 1983.
- [127] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986.
- [128] L. G. Hee, Jr. Ang, and H. Marcelo. *Mobile Robots Navigation, Mapping, and Localization Part I*, pages 1072–1079. 2009. doi : doi:10.4018/978-1-59904-849-9.ch158.

- [129] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3) :338–353, 1965.
- [130] Danica Janglová. Neural networks in mobile robot motion. *International Journal of Advanced Robotic Systems*, 1(1) :2, 2004.
- [131] Hans J Bremermann. *The evolution of intelligence : The nervous system as a model of its environment*. University of Washington, Department of Mathematics, 1958.
- [132] Marco Dorigo and Luca Maria Gambardella. Ant colony system : a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1) :53–66, 1997.
- [133] Anish Pandey, S Pandey, and DR Parhi. Mobile robot navigation and obstacle avoidance techniques : A review. *Int Rob Auto J*, 2(3) :00022, 2017.
- [134] S. Khatoon and Ibraheem. Autonomous mobile robot navigation by combining local and global techniques. *International Journal of Computer Applications*, 37(3) :0975–8887, 2012.
- [135] B.K. Patle, A. Pandey, D. Parhi, A. Jagadeesh, et al. A review : On path planning strategies for navigation of mobile robot. *Defence Technology*, 2019.
- [136] Robin R. Murphy. *Introduction to AI Robotics*. MIT Press, Cambridge, MA, USA, 1st edition, 2000. ISBN 0262133830.
- [137] Igor Škrjanc and Gregor Klančar. Optimal cooperative collision avoidance between multiple robots based on bernstein–bézier curves. *Robotics and Autonomous systems*, 58(1) :1–9, 2010.
- [138] V. Munoz, A. Ollero, M. Prado, and A. Simon. Mobile robot trajectory planning with dynamic and kinematic constraints. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2802–2807 vol.4, 1994.
- [139] John Canny, Ashutosh Rege, and John Reif. An exact algorithm for kinodynamic planning in the plane. *Discrete & Computational Geometry*, 6(3) :461–484, 1991.
- [140] P. Jacobs and J. Canny. Robust motion planning for mobile robots. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2–7 vol.1, 1990.
- [141] E. Hashemi, M. G. Jadidi, and O. B. Babarsad. Trajectory planning optimization with dynamic modeling of four wheeled omni-directional mobile robots. In *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)*, pages 272–277, 2009.
- [142] S. Liu and D. Sun. Optimal motion planning of a mobile robot with minimum energy consumption. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 43–48, 2011.
- [143] V. Delsart, T. Fraichard, and L. Martinez. Real-time trajectory generation for car-like vehicles navigating dynamic environments. In *2009 IEEE International Conference on Robotics and Automation*, pages 3401–3406, 2009.
- [144] T. Suzuki and K. Ohnishi. Trajectory planning of biped robot with two kinds of inverted pendulums. In *2006 12th International Power Electronics and Motion Control Conference*, pages 396–401, 2006.
- [145] M. Brezak and I. Petrović. Time-optimal trajectory planning along predefined path for mobile robots with velocity and acceleration constraints. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 942–947, 2011.
- [146] Chad Goerzen, Zhaodan Kong, and Bernard Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4) :65, 2010.
- [147] Xiang Li and Andreas Zell. Motion control of an omnidirectional mobile robot. In *Informatics in Control, Automation and Robotics*, pages 181–193. Springer, 2009.

- [148] Farhan A Salem. Dynamic and kinematic models and control for differential drive mobile robots. *International Journal of Current Engineering and Technology*, 3(2) :253–263, 2013.
- [149] T. Fukao, H. Nakagawa, and N. Adachi. Adaptive tracking control of a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 16(5) :609–615, 2000.
- [150] Z. Hou, A. Zou, L. Cheng, and M. Tan. Adaptive control of an electrically driven nonholonomic mobile robot via backstepping and fuzzy approach. *IEEE Transactions on Control Systems Technology*, 17(4) :803–815, 2009.
- [151] R. Fierro and F. L. Lewis. Control of a nonholonomic mobile robot : backstepping kinematics into dynamics. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 4, pages 3805–3810 vol.4, 1995.
- [152] X. Yun and Y. Yamamoto. On feedback linearization of mobile robots. 1992.
- [153] Y. Yamamoto and X. Yun. Coordinating locomotion and manipulation of a mobile manipulator. *IEEE Transactions on Automatic Control*, 39(6) :1326–1332, 1994.
- [154] Nilanjan Sarkar, Xiaoping Yun, and Vijay Kumar. Control of mechanical systems with rolling constraints : Application to dynamic control of mobile robots. *The International Journal of Robotics Research*, 13(1) :55–69, 1994.
- [155] X. Yun and Y. Yamamoto. Internal dynamics of a wheeled mobile robot. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, volume 2, pages 1288–1294 vol.2, 1993.
- [156] R. M. DeSantis. Modeling and path-tracking control of a mobile wheeled robot with a differential drive. *Robotica*, 13(4) :401–410, 1995. doi : 10.1017/S026357470001883X.
- [157] A. Albagul and Wahyudi. Dynamic modeling and adaptive traction control for mobile robots. In *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, volume 1, pages 614–620 Vol. 1, 2004.
- [158] K. Thanjavur and R. Rajagopalan. Ease of dynamic modelling of wheeled mobile robots (wmrs) using kane's approach. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 2926–2931 vol.4, 1997.
- [159] Theo JA de Vries, C v Heteren, and Louis Huttenhuis. Modeling and control of a fast moving, highly maneuverable wheelchair. In *Proceedings of the International Biomechatronics workshop*, pages 110–115, 1999.
- [160] Rached Dhaouadi and A Abu Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies : A unified framework. *Advances in Robotics & Automation*, 2(2) :1–7, 2013.
- [161] K. Thanjavur and R. Rajagopalan. Ease of dynamic modelling of wheeled mobile robots (wmrs) using kane's approach. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 2926–2931 vol.4, 1997.
- [162] P. Morin et C. Samson. Commande. *Cours, INRIA*, 2004.
- [163] Sung-On Lee, Young-Jo Cho, Myung Hwang-Bo, Bum-Jae You, and Sang-Rok Oh. A stable target-tracking control for unicycle mobile robots. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 3, pages 1822–1827 vol.3, 2000.
- [164] C Canudasde Wit and OJ Sordalen. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE transaction on Automatic Control*, 37 :1791–1797, 1992.
- [165] Claude Samson. Time-varying feedback stabilization of nonholonomic car-like mobile robots. 1991.
- [166] Ti-Chung Lee, Kai-Tai Song, Ching-Hung Lee, and Ching-Cheng Teng. Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *IEEE Transactions on Control Systems Technology*, 9(2) :305–318, 2001.

- [167] T. H. A. van den Broek, N. van de Wouw, and H. Nijmeijer. Formation control of unicycle mobile robots : a virtual structure approach. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8328–8333, 2009.
- [168] D. Buccieri, D. Perritaz, P. Mullhaupt, Z. Jiang, and D. Bonvin. Velocity-scheduling control for a unicycle mobile robot : Theory and experiments. *IEEE Transactions on Robotics*, 25(2) : 451–458, 2009.
- [169] Khac Duc Do. Formation tracking control of unicycle-type mobile robots with limited sensing ranges. *IEEE Transactions on Control Systems Technology*, 16(3) :527–538, 2008.
- [170] Khac Duc Do and Jie Pan. Nonlinear formation control of unicycle-type mobile robots. *Robotics and Autonomous Systems*, 55(3) :191–204, 2007.
- [171] P. Corke. *Robotics, Vision and Control - Fundamental Algorithms in MATLAB*. Springer, 2011.
- [172] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.
- [173] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises, 4th Edition*. Wiley, 2012.
- [174] M.R. Rajamani B.J. Odelson and J.B. Rawlings. A new autocovariance least-squares method for estimating noise covariances. 2003.
- [175] S.C. Rutan. Adaptive kalman filtering. *Analytical Chemistry*, 1991.
- [176] O. Kost, O. Straka, and J. Dunik. Identification of state and measurement noise covariance matrices using nonlinear estimation framework. 659 :012057, 11 2015.
- [177] Granjon Yves. Automatique, systèmes linéaires, non linéaires, à temps continu, à temps discret, représentation d'état. *Cours, Dunod, Paris*, 2010.
- [178] Keisuke Fujii. Extended kalman filter. *Refernce Manual*, pages 14–22, 2013.
- [179] Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2) :229–252, dec 2008.
- [180] R. J. Patton. Fault-tolerant control systems : The 1997 situation. In *IFAC SAFEPROCESS*, pages 1033–1054, 1997.
- [181] J. Jiang and Y. Xiang. Fault-tolerant control systems : A comparative study between active and passive approaches. 36 :60–72, 04 2012.
- [182] R. J. Patton. *Fault-Tolerant Control*, pages 1–8. Springer London, London, 2014.
- [183] An-Min Zou and Krishna Dev Kumar. Adaptive fuzzy fault-tolerant attitude control of spacecraft. *Control Engineering Practice*, 19(1) :10–21, 2011.
- [184] Mogens Blanke, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki, and Jochen Schröder. *Diagnosis and fault-tolerant control*, volume 2. Springer, 2006.
- [185] Montadher Sami and Ron J Patton. Active fault tolerant control for nonlinear systems with simultaneous actuator and sensor faults. *International Journal of Control, Automation and Systems*, 11(6) :1149–1161, 2013.
- [186] Jan Lunze and Thomas Steffen. Control reconfiguration after actuator failures using disturbance decoupling methods. *IEEE Transactions on Automatic Control*, 51(10) :1590–1601, 2006.
- [187] Patrick F Muir and Charles P Neuman. Kinematic modeling of wheeled mobile robots. *Journal of robotic systems*, 4(2) :281–340, 1987.
- [188] Hamid Taheri, Bing Qiao, and Nurallah Ghaeminezhad. Kinematic model of a four mecanum wheeled mobile robot. *International journal of computer applications*, 113(3) :6–9, 2015.

- [189] Pakpoom Viboonchaicheep, Akira Shimada, and Yuhki Kosaka. Position rectification control for mecanum wheeled omni-directional vehicles. In *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No. 03CH37468)*, volume 1, pages 854–859. IEEE, 2003.
- [190] Ching-Chih Tsai and Hsiao-Lang Wu. Nonsingular terminal sliding control using fuzzy wavelet networks for mecanum wheeled omni-directional vehicles. In *International Conference on Fuzzy Systems*, pages 1–6. IEEE, 2010.
- [191] Lih-Chang Lin and Hao-Yin Shih. Modeling and adaptive control of an omni-mecanum-wheeled robot. 2013.
- [192] Ching-Chih Tsai, Feng-Chun Tai, and Ying-Ru Lee. Motion controller design and embedded realization for mecanum wheeled omnidirectional robots. In *2011 9th World Congress on intelligent control and automation*, pages 546–551. IEEE, 2011.
- [193] Veer Alakshendra and Shital S Chiddarwar. Adaptive robust control of mecanum-wheeled mobile robot with uncertainties. *Nonlinear Dynamics*, 87(4) :2147–2169, 2017.
- [194] Veer Alakshendra and Shital S Chiddarwar. A robust adaptive control of mecanum wheel mobile robot : simulation and experimental validation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5606–5611. IEEE, 2016.
- [195] Veer Alakshendra, Shital S. Chiddarwar, Gajanan Nikhade, Abhishek Jha, and Saumya Ranjan Sahoo. Robust trajectory tracking control for an omnidirectional mobile robot. In *Proceedings of the Advances in Robotics*, AIR '17, New York, NY, USA, 2017. Association for Computing Machinery. doi : 10.1145/3132446.3134884. URL <https://doi.org/10.1145/3132446.3134884>.
- [196] Saumya R Sahoo, Shital S Chiddarwar, and Veer Alakshendra. Intuitive dynamic modeling and flatness-based nonlinear control of a mobile robot. *Simulation*, 94(9) :797–820, 2018.
- [197] A. Isidori and C. De Persis. Feedback linearization of nonlinear systems. *Control Systems, Robotics and AutomatioN–Volume XII : Nonlinear, Distributed, and Time Delay Systems-I*, page 193, 2009.
- [198] D. Ichalal, B. Marx, J. Ragot, and D. Maquin. Unknown input observer for lpv systems with parameter varying output equation. *IFAC-PapersOnLine*, 48(21) :1030–1035, 2015.
- [199] G. C.Karras and G. K. Fourlas. Model predictive fault tolerant control for omni-directional mobile robots. *Journal of Intelligent & Robotic Systems*, 2019. doi : <https://doi.org/10.1007/s10846-019-01029-7>.
- [200] S. Mishra, M. Sharma, and S. Mohan. Behavioural fault tolerant control of an omni directional mobile robot with four mecanum wheels. *Defence Science Journal*, 69(4) :353–360, Jul. 2019. doi : 10.14429/dsj.69.13607. URL <https://publications.drdo.gov.in/ojs/index.php/dsj/article/view/13607>.
- [201] Jung-Hoon Hwang, R. C. Arkin, and Dong-Soo Kwon. Mobile robots at your fingertip : Bezier curve on-line trajectory generation for supervisory control. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Cat. No.03CH37453)*, volume 2, pages 1444–1449 vol.2, 2003.
- [202] T. M. Howard and A. Kelly. Trajectory and spline generation for all-wheel steering mobile robots. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4827–4832, 2006.

Abstract

The work presented in this thesis is a part of the European project *PRODUCTIVE4.0* in partnership with *STMicroelectronics* of Rousset, France. In order to optimize the production, *STMicroelectronics* integrates two types of wheeled mobile robots (unicycle mobile robots and omnidirectional mobile robots with 4 Mecanum wheels) in the fab (semiconductor manufacturing facility) to transport products between the various equipments in the production chain. In this thesis, we distinguish two major issues that can prevent the robots from well performing their task : the apparition of *faults* at the actuators or the sensors level, and the delay that can be generated by moving and unpredictable obstacles while navigating.

Firstly, we are interested in unicycle robots. We propose a Kalman observer-based approach to diagnose the actuators and sensors *faults*. Then, an approach based on the *faults* estimation and compensation allows to accommodate these faults. Secondly, we focus on omnidirectional robots and we propose methods based on observers (Kalman and unknown inputs) in order to diagnose the actuators and sensors *faults*. After that, the impact of the actuators *fauts* on the robots behavior is studied. Finally, we deal with the problem linked to the obstacles generated delay by proposing a methodology based on the reference trajectories reconfiguration in order to compensate this delay.

Résumé

Ce travail de thèse s'inscrit dans le cadre du projet Européen *PRODUCTIVE4.0* en partenariat avec *STMicroelectronics* de Rousset, France. Afin d'optimiser la production, *STMicroelectronics* intègre deux types de robots mobiles à roues (les robots unicycles et les robots omnidirectionnels à 4 roues Mecanum) dans la fab (installations de fabrication du semi-conducteur) pour le transport des produits entre les divers équipements de la chaîne de production. Dans le cadre des travaux de cette thèse, nous distinguons deux problématiques majeures pouvant empêcher les robots de bien effectuer leur tâche : l'apparition de *défauts* au niveau des actionneurs ou des capteurs, et le retard qui peut être généré par les obstacles mobiles et imprévisibles lors de la navigation des robots.

Nous nous intéressons dans un premier temps aux robots unicycles. Nous proposons une approche à base d'observateur de Kalman afin de diagnostiquer les *défauts* actionneurs et capteurs. Une approche basée sur l'estimation et la compensation des *défauts* permet ensuite de les accomoder. Dans un second temps, nous nous focalisons sur les robots omnidirectionnels et nous proposons des méthodes à base d'observateurs (de Kalman et à entrées inconnues) afin de diagnostiquer les *défauts* actionneurs et capteurs. Ensuite, l'impact des *défauts* actionneurs sur le comportement des robots est étudié. Enfin, nous traitons la problématique liée au retard généré par les obstacles en proposant une méthodologie basée sur la reconfiguration des trajectoires de référence afin de compenser ce retard.