

# Diagnostic de plateformes robotisées. Méthode hybride évolutive intégrant une interaction avec un opérateur expert.

Mahmoud Almasri

### ► To cite this version:

Mahmoud Almasri. Diagnostic de plateformes robotisées. Méthode hybride évolutive intégrant une interaction avec un opérateur expert.. Automatique / Robotique. Université Clermont Auvergne [2017-2020], 2020. Français. NNT: 2020CLFAC035. tel-03065872

## HAL Id: tel-03065872 https://theses.hal.science/tel-03065872

Submitted on 15 Dec 2020  $\,$ 

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés. Université Clermont Auvergne Ecole Doctorale Sciences Pour l'Ingénieur

# Thèse Présentée par Mahmoud Almasri

Irstea

Pour obtenir le grade de Docteur d'Université

Spécialité:

Diagnostic de plateformes robotisées

Méthode hybride évolutive intégrant une interaction avec un opérateur expert

Soutenue publiquement le 20 février 2020 devant le Jury

М.	Jean-Christophe	Popieul	Président
М.	Maan	El Badaoui El Najjar	Rapporteur
М.	Kouider Nacer	M'Sirdi	Rapporteur
Mme.	Karen	Godary-Dejean	Examinatrice
М.	Benoît	Thuilot	Examinateur
М.	Nicolas	Tricot	Encadrant de thèse
М.	Roland	LENAIN	Directeur de thèse

 $\dot{A}$  ma famille

## Résumé

La robotique occupe une place grandissante dans des secteurs très variés allant de l'agriculture à l'industrie. Son développement s'est vu favorisé par l'intérêt pour l'Homme de disposer d'entités autonomes capables de l'aider pour la réalisation d'actions difficiles voire impossibles, soit de par leur pénibilité, soit de par leur dangerosité.

La robotique mobile, l'une des filières de la robotique, poursuit cet objectif grâce aux avancées technologiques et scientifiques dans des domaines tels que la mécanique, l'électronique, l'automatique et l'informatique. Ces robots évoluent de manière autonome dans des environnements complexes et sur de longues durées. Leurs algorithmes garantissent une certaine robustesse face à des situations prévues. Cependant, dans certaines situations (dysfonctionnement d'un composant physique du robot, absence d'un signal GPS dans le cadre d'une tâche de suivi de trajectoire, ...), l'apparition d'un défaut rendra impossible la tâche qui incombait au robot.

Afin d'éviter ce type de situation, le diagnostic en temps réel des défauts d'un robot intégrant dans sa boucle de détection comme de proposition de solutions un opérateur humain apparait être une solution possible et potentiellement efficace. Les travaux décrits dans ce mémoire visent cet objectif. Ils proposent, d'une part, un système de supervision du diagnostic de défauts d'un robot mobile et, d'autre part, un ensemble d'outils de modélisation, de paramétrage et d'adaptation des méthodes de diagnostic aux défauts potentiels d'un robot.

La première partie des contributions est composée de deux modules. Le premier module présente une méthode hybride de diagnostic mettant en œuvre plusieurs approches et types d'approche de diagnostic de défauts. Cette méthode permet de diagnostiquer une liste définie de défauts en temps réel. Le second décrit un module d'interaction homme/robot permettant l'intégration de l'opérateur dans la boucle de diagnostic en proposant/validant des solutions mais aussi en corrigeant, si nécessaire, les décisions prises par la méthode hybride proposée. Ce module intègre une base de connaissances et un système de raisonnement à partir de cas permettant de conserver l'historique des décisions prises par l'opérateur, considéré expert, afin d'améliorer sa performance en continu. La deuxième partie détaille la démarche amont et hors ligne de la mise en œuvre de la méthode hybride de diagnostic. Cette démarche comprend les étapes de caractérisation des défauts considérés, de définition et d'adaptation des méthodes de diagnostic capables de diagnostiquer les défauts identifiés. Cette adaptation consiste à définir les modèles, à paramétrer les méthodes et, dans certains cas, à recueillir des données d'entraînement.

Du point de vue expérimental, les développements théoriques sont validés sur des données issues d'un robot réel sans train directeur et de simulations.

Littérairement, ce mémoire détaille ces travaux en cinq chapitres. Le premier chapitre introduit le contexte de la thèse. Le second définit la problématique, les hypothèses proposées et décrit l'état de la littérature sur les méthodes de diagnostic de défauts. Les développements théoriques portant sur le système de supervision du diagnostic de défauts robot et sur les outils aboutissant à son paramétrage sont décrits dans les deux chapitres qui suivent. Les résultats de la mise en œuvre des développements théoriques en simulation et sur un cas réel de robot sont montrés et discutés dans le cinquième chapitre. Enfin, une conclusion et les perspectives envisageables de ces travaux sont présentées.

## Abstract

Robotics is becoming increasingly important in a wide variety of sectors. Its development has been favored by the interest for Man to have autonomous entities capable of helping him to perform actions that are difficult or impossible for human being, either because of their painfulness or because of their dangerousness. The Mobile robotics, one of the sectors of robotics, pursues this objective thanks to technological and scientific advances in different areas such as mechanics, electronics, automation and computer science. These robots navigate autonomously in complex environments and over long periods of time. Their algorithms guarantee a certain robustness while dealing with foreseen situations.

However, in some situations (malfunctioning of a physical component of the robot, loss in GPS signal while executing a trajectory tracking task, etc.), the appearance of a fault may cause the robot to fail in its mission.

In order to avoid this type of situation, real-time fault diagnosis integrating a human operator in its fault detection and fault accommodation loops appears to be a possible and potentially effective solution. The work described is aimed at this objective. It offers, on the one hand, a fault diagnosis supervision system of a mobile robot and, on the other hand, a set of tools for modeling, parameterizing and applying for diagnosis methods to potential fault of a robot.

The first part of the contributions is composed of two modules. The first module is a hybrid diagnostic method that implements several approaches and types of fault diagnosis approach. This method can diagnose a defined list of faults in real time. The second is a human/robot interaction module allowing the integration of the operator in the diagnostic loop by proposing/validating solutions and by correcting, if necessary, the decisions made by the proposed hybrid method. This module integrates a knowledge database and a case-based reasoning algorithm allowing to save the history of the decisions made by the operator, considered expert, in order to improve his performance continuously.

The second part details the approach done off-line and upstream to the implementation of the hybrid diagnostic method. This approach includes the characterization of the considered faults, the definition and the adaptation of the diagnosis methods making it possible to diagnose the identified faults. The adaptation consists of defining the models, adjusting the parameters and, in some cases, collecting training data. From experimental point of view, the theoretical developments are validated on data coming from a real skid-steering robot and from simulations.

Literally, this document details these works in five chapters. The first chapter introduces the context of the thesis. The second defines the problem statement, the proposed hypotheses and describes the state of the literature of fault diagnosis methods. Theoretical developments concerning fault diagnosis supervision system of mobile robotics and the tools leading to its configuration are described in two chapters that follow. The application results of the theoretical developments in simulation and on a real robot case are shown and discussed in the fifth chapter. Finally, a conclusion and the possible perspectives of this work are presented.

### Remerciements

Cette thèse a été réalisée à l'Institut National de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture (IRSTEA) de Clermont-Ferrand, dirigé par Emmanuel HUGO, au sein de l'équipe de l'unité de recherche technologies et systèmes d'information pour les agrosystèmes (TSCF) dirigé par Jean-Pierre CHANET. Ella a été financée par le FEDER (Fonds Européens de Développement Régional) et la région Auvergne-Rhône-Alpes.

J'adresse donc d'abord mes remerciements à l'ensemble des représentants de ces établissements et à leurs personnels pour leur accueil ainsi que pour l'ensemble des moyens mis à ma disposition durant cette période de thèse.

Je souhaite ensuite remercier les membres du jury, qui ont eu l'amabilité d'examiner mes travaux. Merci en particulier à Maan EL BADAOUI EL NAJJAR, professeur à l'Université de Lille qui m'a initié au monde de la recherche, Kouider Nacer M'Sirdi professeur à l'Université Aix-Marseille et Karen GODARY-DEJEAN maître de conférences à l'Université de Montpellier d'avoir assisté à mes comités de thèses.

J'adresse également de vifs remerciements à Roland LENAIN, mon directeur de thèse et l'animateur de l'équipe RObotique et Mobilité pour l'Environnement et l'Agriculture (ROMEA) pour ses remarques pertinentes et ses précieux conseils.

Je tiens à exprimer mes sincères remerciements à Nicolas TRICOT, mon co-encadrant de thèse, pour sa disponibilité, les relectures assidues et les réunions hebdomadaires constructives tout au long de ma thèse.

Je ne peux pas oublier Adrian COUVENT et Guillaume PICARD, mes chers collègues de bureau, mais avant tout mes amis. Un grand merci pour l'ambiance magnifique de notre bureau et pour leurs discussions en robotique et en philosophie! Bonne chance dans vos thèses.

Merci également aux membres de l'équipe ROMEA avec qui j'ai passé d'agréables moments.

Finalement, je tiens à remercier chaleureusement ma famille, ma mère, mon père et mes frères et sœurs qui me soutiennent depuis toujours. Je remercie du fond du cœur ma femme, Hana, pour l'amour indéfectible et le soutien immuable qu'elle a su me témoigner. Sans elle rien n'aurait été possible.

# Liste Des Abbrévations

AGV	Automatic Guided Vehicle (véhicule à guidage automatique)
Bdc	Base de Connaissance
С	Classificateur
CBR	Case-Based Reasoning (raisonnement à partir de cas)
DBN	Deep Belief Network
DD	Délai de Détection
DL	Deep-Learning (apprentissage profond)
EC	Encodeur et Classificateur
EE	Estimation d'Erreur
EKF	Extended Kalman Filter (filtre de Kalman étendu)
FAR	False Alarm Rate (taux de fausses alarmes)
FDI	False Isolation Rate (taux de faux isolements)
GPF	Gaussian Particle Filter (filtre à particules gaussien)
ID	Isolation Delay (délai d'isolement)
IHR	Interface Homme/Robot
IMU	Inertial Measurement Unit (Centrale à inertie)
KF	Kalman Filter
LDA	Linear Discriminant Analysis (Analyse discriminante linéaire)
LSTM	Long Short-Term Memory
MDR	Missed Detection Rate (taux de détections ratées)
$MHD_n$	Méthode Hybride de Diagnostic - énième forme
MIHR	Module d'Interaction Homme/Robot
MMAE	Multiple-Model Adaptive Estimation (estimation adaptative multi-modèles)
NMRD	Nominal Mode Return Delay (Délai de rétablissement di mode normal)
OMOPAMED2	Outils de Modélisation, de Paramétrage et d'Adaptation des MÉthodes de
	Diagnostic aux Defauts
OPAMED2	Outlis de Parametrage et d'Adaptation des MEthodes de Diagnostic aux Defauts
PF	Particle Filter (filtre a particules)
PFM	Filtre a Particules Mixe
PP DDDE	Pose Provider (lournisseur de position)
RBPF	Filtre a Particules Rao-Blackwell
RDS	Required Data Storage (volume de stockage requis)
RLS	Recursive Least square (algorithme des moindres carres recursifs)
RPE	Recursive Parameter Estimation (estimation recursive des parametres)
RSPF	Risk Sensitive Particle Fliter (Flitre a particules sensible aux risques)
S2D2R	Systeme de Supervision Des Defaut Robots
TPC	Time Per Cycle (temps de calcul par cycle)
UKF	Unscented Kalman Filter
UPF'	Unscented Particle Filter
UWB	Ultra WideBand
VRPF	Variable Resolution Particle Filter (filre à particule à résolution variable)

# Table des matières

1	Intr	roduction	3
	1.1	La robotique et la nécessité du diagnostic de défauts	3
		1.1.1 Robots manipulateurs : un type répandu de robots	3
		1.1.2 Robots mobiles en milieu maîtrisé	3
		1.1.3 Robots dans des milieux plus divers	3
	1.2	Contexte scientifique et applicatif	4
	1.3	Objectifs	6
<b>2</b>	Pro	blématique du diagnostic des défauts en robotique mobile	8
	2.1	Importance du diagnostic de défauts en robotique mobile	8
	2.2	Définitions	8
		2.2.1 Défauts	8
		2.2.2 Diagnostic des défauts	10
	2.3	Problématique	10
		2.3.1 Cahier des charges pour les méthodes de diagnostic	10
		2.3.2 Difficultés	11
		2.3.3 Hypothèses proposées	12
		2.3.4 Méthodologie de recherche	13
	2.4	Généralités sur les méthodes de diagnostic	14
	2.5	Classification des méthodes répondant au cahier des charges	18
	2.6	Attribution des méthodes de diagnostic aux défauts	18
	2.7	L'importance de l'humain	23
	2.8	Les contributions de la thèse	23
3	Mét	thodologie de conception d'un système homme/robot de diagnostic de ro-	
	$\mathbf{bot}$	mobile	25
	3.1	Schéma global de la méthodologie	25
	3.2	Proposition d'une Méthode Hybride de Diagnostic : la MHD	26
		3.2.1 Les étapes de la MHD	26
		3.2.2 Évaluation des méthodes de diagnostic	27
	3.3	Proposition d'un module d'interaction homme robot	30
		3.3.1 Défaut isolé	31
		3.3.2 Défaut détecté non isolé	37
		3.3.3 Ajustement des sous-méthodes	37
		3.3.4 Défaut signalé par l'opérateur	38
		3.3.5 L'interaction avec l'humain	38

	ustement et optimisation des methodes de diagnostic de defauts en rot	0
/ 1	Les filtres de Kalman	
7.1	4.1.1 Définitions	•
	4.1.1 Demittons	•
	4.1.2 Auto-parametrage des intres de Rainian	•
	4.1.5 Diagnostic de défauts les capteurs de localisation	•
19	Los Bolations de redendance	•
4.2	L'estimation réquirire de paramètres	•
4.5	Multiple Model Adentive Estimation (MMAE)	•
4.4	Eltrog à particulas (DE)	•
4.0	Fittees a particules (PF) $\dots$	•
	4.5.1 PF Standard $\dots$	·
	4.5.2 Filtre a particules sensible aux risques	•
	4.5.3 Filtre a particules a resolution variable	•
	4.5.4 Filtre a particules Rao-Blackwell	•
	4.5.5 Filtre a particules Gaussien et Filtre a particules Unscented	•
1.0	4.5.6 Proposition d'un filtre à particules : Filtre à Particules Mixe	•
16	L'apprentissage profond	
4.0		
4.0	4.6.1 Proposition de modèles basés sur l'apprentissage profond	
4.0 4.7	4.6.1 Proposition de modèles basés sur l'apprentissage profond Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur	r
4.0	4.6.1 Proposition de modèles basés sur l'apprentissage profond Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance	r
<ul><li>4.0</li><li>4.7</li><li>4.8</li></ul>	4.6.1 Proposition de modèles basés sur l'apprentissage profond Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance	r
<ul><li>4.0</li><li>4.7</li><li>4.8</li><li>Mit</li></ul>	4.6.1 Proposition de modèles basés sur l'apprentissage profond Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit 5.1</li> </ul>	4.6.1 Proposition de modèles basés sur l'apprentissage profond Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit 5.1</li> </ul>	<ul> <li>4.6.1 Proposition de modèles basés sur l'apprentissage profond Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance</li></ul>	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit</li> <li>5.1</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur         le vecteur de confiance         Conclusion         Conclusion         de la démarche de diagnostic proposée         Définition du contexte d'application de la démarche         5.1.1         Environnement d'évaluation         5.1.2         Définition du Robot	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit</li> <li>5.1</li> </ul>	<ul> <li>4.6.1 Proposition de modèles basés sur l'apprentissage profond Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance</li></ul>	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit</li> <li>5.1</li> <li>5.2</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond          Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance          Conclusion           Conclusion           Se en application de la démarche de diagnostic proposée          Définition du contexte d'application de la démarche          5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot          5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit 5.1</li> <li>5.2</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance         Conclusion         Conclusion         See en application de la démarche de diagnostic proposée         Définition du contexte d'application de la démarche         5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot         5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2       Service         5.2.1       Identification des défauts	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit</li> <li>5.1</li> <li>5.2</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond          Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit 5.1</li> <li>5.2</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance         Conclusion         Conclusion <b>See en application de la démarche de diagnostic proposée</b> Définition du contexte d'application de la démarche         5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot         5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2         5.2.1       Identification des défauts         5.2.2       Couples défauts/méthodes	r
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit</li> <li>5.1</li> <li>5.2</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur         le vecteur de confiance         Conclusion         Conclusion         Se en application de la démarche de diagnostic proposée         Définition du contexte d'application de la démarche         5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot         5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2         5.2.1       Identification des défauts         5.2.3       Mise en œuvre d'OPAMED2         5.2.4       Modèle du robot et des défauts	
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit 5.1</li> <li>5.2</li> <li>5.3</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance         Conclusion         Gonduision         Se en application de la démarche de diagnostic proposée         Définition du contexte d'application de la démarche         5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot         5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2         5.2.2       Couples défauts/méthodes         5.2.3       Mise en œuvre d'OPAMED2         5.2.4       Modèle du robot et des défauts	. r 
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mii</li> <li>5.1</li> <li>5.2</li> <li>5.3</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance         Conclusion         Se en application de la démarche de diagnostic proposée         Définition du contexte d'application de la démarche         5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot         5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2         5.2.2       Couples défauts/méthodes         5.2.3       Mise en œuvre d'OPAMED2         5.2.4       Modèle du robot et des défauts         5.3.1       Définition de la version initiale de la MHD	. r 
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mit 5.1</li> <li>5.2</li> <li>5.3</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance         Conclusion         Se en application de la démarche de diagnostic proposée         Définition du contexte d'application de la démarche         5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot         5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2         5.2.1       Identification des défauts         5.2.2       Couples défauts/méthodes         5.2.3       Mise en œuvre d'OPAMED2         5.2.4       Modèle du robot et des défauts         5.3.1       Définition de la version initiale de la MHD	· · · · · · · · · · · · · · · · · · ·
<ul> <li>4.0</li> <li>4.7</li> <li>4.8</li> <li>Mii</li> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> </ul>	4.6.1       Proposition de modèles basés sur l'apprentissage profond         Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur         le vecteur de confiance         Conclusion         Conclusion         Se en application de la démarche de diagnostic proposée         Définition du contexte d'application de la démarche         5.1.1       Environnement d'évaluation         5.1.2       Définition du Robot         5.1.3       Caractéristiques de l'opérateur         Mise en œuvre d'OMOPAMED2         5.2.1       Identification des défauts         5.2.2       Couples défauts/méthodes         5.2.3       Mise en œuvre d'OPAMED2         5.2.4       Modèle du robot et des défauts         5.3.1       Définition de la version initiale de la MHD         5.3.2       Mise en œuvre du MIHR         S.3.2       Mise en œuvre du MIHR	· · · · · · · · · · · · · · · · · · ·

2

# Chapitre 1

# Introduction

### 1.1 La robotique et la nécessité du diagnostic de défauts

La robotique est un domaine très actif actuellement. Son développement est accéléré grâce aux rôles importants que jouent ces machines dans différents domaines. Les tâches fastidieuses, pénibles ou dangereuses y sont de plus en plus confiées. Selon [31] le terme "robot" est défini comme une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a. Ainsi, en respectant cette définition, plusieurs types de robots existent.

### 1.1.1 Robots manipulateurs : un type répandu de robots

Depuis les années 1960, les robots sont présents sur les chaînes de production dans les usines. Les tâches à effectuer sont souvent répétitives et elles se déroulent dans un milieu maîtrisées. Les travaux peuvent être confiés aux bras robotisés qui font en permanence les mêmes trajectoires afin d'effectuer les tâches [41].

### 1.1.2 Robots mobiles en milieu maîtrisé

Un robot mobile est un robot capable de locomotion. Plusieurs moyens de locomotion existent dans ce domaine comme les pattes [37], le glissement [79] les chenilles et les roues. Dans les usines, les robots à roues ou à chenilles assurent le transport des marchandises [65]. Des techniques classiques comme le filoguidage [65] permettent aux robots de naviguer dans ces milieux.

### 1.1.3 Robots dans des milieux plus divers

Les avancées dans les différents domaines intégrant la robotique ont donné aux robots des capacités d'adaptation poussées permettant à ces machines d'évoluer dans des milieux plus divers. Pour réussir leurs déplacements dans des milieux incertains, ces entités multi-capteurs se sont vus dotés de diverses fonctionnalités.

**Perception** Le robot mise sur ces capteurs pour "voir" et avoir connaissance de l'environnement qui l'entoure. Selon le contexte, différents capteurs peuvent être associés au robot pour réussir la perception de l'environnement comme les caméras [65], les radars, les télémètres lasers (2D [46] et 3D [16]), etc.

**Localisation** Pour réussir sa mission, le robot doit connaître, à tout moment, sa position absolue (dans une carte connue) ou relative (par rapport à des objets connus). Il existe une vaste liste de capteurs qui peuvent contribuer à la localisation. Par exemple, la fusion des vitesses des roues (issues des encodeurs) et des informations issues d'une caméra de profondeur (Kinect) [6] ou, la fusion des informations issues des encodeurs, d'une centrale inertielle et d'un radar Doppler pour une localisation dans un environnement complexe et dynamique [55].

**Guidage et contrôle** En s'appuyant sur les informations issues de la perception et de la localisation, un robot autonome planifie son chemin et suit sa trajectoire grâce aux algorithmes de guidage et de contrôle. Les techniques de contrôle et de guidage sont nombreuses. Par exemple, le contrôle adaptatif et prédictif est utile pour le suivi de trajectoire à haute précision dans des milieux naturels [50]. Le contrôle par retour d'état est puissant dans le cas des systèmes linéaires en sachant que beaucoup des systèmes non linéaires sont représentables d'une façon linéaire suite à une phase supplémentaire de linéarisation [12]. Le Backstepping est une technique générique de contrôle compatible avec la représentation non linéaire [24]. Des applications comme le suivi de structures [24], le suivi de trajectoires [50], la formation de flottes des robots [51] sont rendues possibles grâce aux diverses techniques de commande.

Toutes ces variabilités impliquent des risques de problèmes plus ou moins graves liés au fonctionnement des divers composants du robot, à son interaction avec l'environnement et aux algorithmes implémentés. En effet, pour exécuter leurs missions, les robots sont équipés des différents composants (capteurs, actionneurs, éléments mécaniques) et seront amenés à se trouver dans des contextes différents ce qui ouvre les portes à la présence de défaut issus de différentes types (nombreux composants du robot, liés à l'interaction avec l'environnement, ...). Ne pas prendre en compte les défauts lors de la conception et du déploiement du robot amènera à des conséquences catastrophiques sur le robot et l'environnement dans lequel il se trouve (robot hors de contrôle, dommage des matériels et des personnes...). Le diagnostic de défauts est donc une tâche importante qui ne doit pas être négligée. Cette thèse aborde la problématique du diagnostic des défauts des robots mobiles.

### 1.2 Contexte scientifique et applicatif

La thèse s'inscrit dans le cadre du projet RTIF "Robotique et Transitique Industrielle Fiable", financé par la région Auvergne-Rhône-Alpes en lien avec le Fond Européen de Développement Régional (FEDER). Ce projet vise à traiter les différents problèmes que posent conjointement l'usine du futur et l'agriculture du futur, en termes de fiabilité des produits et de robustesse des procédés. Il porte plus particulièrement sur la combinaison de la conception et de la flexibilité du pilotage d'un atelier ou d'une exploitation agricole fortement robotisée. Il contient trois axes principaux : l'anticipation, le pilotage a priori et a postériori des opérations, c'est-à-dire la correction de dysfonctionnements constatés. La thèse porte sur ce dernier. Elle s'est déroulée au sein de l'Unité de Recherche "Technologies et systèmes d'information pour les agrosystèmes (TSCF)" à l'Institut national de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture (Irstea). Les deux autres axes sont traités par les angles "Pilotage fiable et robuste de systèmes de production et de transitique" et "Développement des méthodes de propagation d'incertitudes pour la prise de décision en contexte incertain" et se déroulent respectivement au Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS), et à SIGMA-Clermont au sein de l'Institut Pascal dans le cadre de deux thèses.

L'application de cette thèse s'étend au domaine de l'industrie et au domaine de l'agriculture. Les robots mobiles permettent de réaliser des tâches dangereuses, complexes, répétitives et/ou fastidieuses dans un environnement complexe et dynamique. Différents types de robots agricoles



(a) Robot suiveur

(b) Robot porteur enjambeur SITIA



(c) Robot pulvérisateur

(d) Flotte de robots

sont illustrés à la figure 1.1. La figure de gauche (1.1a) illustre un robot suiveur pouvant aider l'agriculteur au transport des matériels ou des fruits récoltés [49]. Un robot enjambeur polyvalent [75] est montré sur la figure 1.1b. Équipé d'un pulvérisateur et des capteurs, le robot présenté sur la figure 1.1c peut assurer une pulvérisation plus précise et une distribution plus homogène des produits [18]. Finalement, une flotte de robots agricoles [51] est illustrée sur la figure 1.1d où deux robots suivent un tracteur.

Ces exemples montrent l'hétérogénéité des robots agricoles. De plus, Ces derniers sont équipés de nombreux capteurs et travaillent dans des contextes différents et des milieux incertains. Par conséquent, de nombreuses sources potentielles de défauts apparaissent.

Les véhicules à guidage automatique "Automatic Guided Vehicles" (AGV) sont répandus dans le contexte industriel. Les technologies de localisation/guidage utilisées peuvent être différentes d'un AGV à l'autre par exemple [65]. Ainsi, certains AGV utilisent le filoguidage, d'autres le

FIGURE 1.1 – Exemples des robots agricoles



FIGURE 1.2 – Exemples des robots industriels

guidage par laser, le guidage par ultra son, le guidage par vision ou encore le géo-guidage... L'architecture mécanique des AGV n'est pas unique également. Il existe des AGV à un train directeur unique, d'autres à deux trains directeurs, équipés ou non d'un différentiel [65]. D'un point de vue modélisation, ces robots peuvent être représentés à l'aide de modèles mathématiques diverses comme les modèles cinématiques ou dynamiques, dans des référentiels abstraits ou reliés aux trajectoires. Plusieurs AGV sont présentés à la figure 1.2. Les deux AGV référencés a et b dans cette figure illustrent deux types de contact robot-sol, l'un avec des roues et deux trains et l'autre avec deux trains de chenilles. Les images c, d et e présentent trois exemples d'application : le levage, le traînage et la manipulation d'objets à l'aide d'un robot équipé d'un bras robotisé. Les images f et g illustrent deux technologies de navigation : la navigation avec un fil magnétique (filoguidage) et la navigation laser en s'appuyant sur des balises pour se localiser (guidage par laser).

On retrouve ici l'hétérogénéité des types de robot relevé dans le contexte agricole. Par conséquent, les sources potentielles des défauts dans le cas des robots mobiles industriels sont nombreuses également.

Donc, que ce soit dans les industries ou dans les champs agricoles, beaucoup de dysfonctionnements potentiels peuvent nuire au fonctionnement des robots mobiles. Ce qui montre que le diagnostic en temps réel des défaut est une tâche cruciale afin de garantir l'efficacité de ces machines autonomes.

# 1.3 Objectifs

L'objectif des travaux de la thèse est d'assurer un fonctionnement fiable des robots. Ceci par le diagnostic de défauts qui peuvent survenir à n'importe quel moment sur un robot mobile. L'idée proposée est de concevoir une méthode de diagnostic qui combine la puissance de l'informatique et l'intelligence humaine. Dans un premier temps, les défauts seront diagnostiqués automatiquement via les algorithmes adaptés puisque l'autonomie des robots passe par une capacité à gérer seul le défaut. Puis, l'humain interviendra soit pour aider l'algorithme à trouver le défaut

s'il en est pas capable automatiquement, soit pour prendre en main le robot et corriger la situation. Plusieurs scénarios de test sont fixés puis les approches sont validées en simulation et sur des données issues du robot réel. Les défauts considérés sont répartis dans trois catégories : dysfonctionnement d'un composant mécanique, perturbation sur le signal d'un capteur et dysfonctionnement causé par des éléments externes par rapport au robot.

Dans la suite de ce document, la problématique de recherche et les hypothèses faites sont détaillées dans le chapitre 2 ainsi qu'un état d'art portant sur les méthodes de diagnostic en robotique mobile. Dans ce chapitre, une liaison entre les méthodes de diagnostic et les défauts considérés dans cette étude est établie. Le chapitre 3 décrit les principaux éléments de la méthode hybride de diagnostic proposée, notamment l'indicateur de performance qui sert à évaluer et à comparer d'une manière fiable les méthodes de diagnostic. Ensuite, les méthodes de diagnostic proposées et les ajustements faits à ces méthodes de façon à pouvoir les appliquer aux défauts identifiés sont expliqués dans le chapitre 4. Finalement, le chapitre 5 résume les résultats obtenus par la mise en œuvre des méthodes de diagnostic. Une forme de la méthode hybride de diagnostic y est déduite en se basant sur l'évaluation des méthodes de diagnostic.

# Chapitre 2

# Problématique du diagnostic des défauts en robotique mobile

## 2.1 Importance du diagnostic de défauts en robotique mobile

La robotique mobile est de plus en plus ancrée dans notre vie quotidienne et ses applications se multiplient. Selon une projection faite par la fédération internationale de robotique parue en 2016 [70], 31 millions de robots de service auraient été vendus dans le monde entre 2014 et 2017. Les robots mobiles autonomes à roues ont de nombreuses applications, comme le service, l'exploration de l'espace [23], le transport des objets [64] et la production et la gestion agricole. Actuellement, les robots agricoles aident à la récolte, à la pulvérisation, à la cueillette et au contrôle de produits chimiques [48].C'est un champ applicatif de plus en plus porteur. En effet, selon un rapport de Tractica [4], les commandes de robots agricoles augmenteront considérablement dans les années à venir, passant de 32 000 unités en 2016 à 594 000 unités par an en 2024, alors que le marché mondial devrait atteindre 74,1 milliards de dollars de revenus annuels.

Ces machines deviennent de plus en plus autonomes et exécutent des tâches pour de longues durées [56]. Cependant, chaque robot mobile rencontrera des défauts au cours de sa vie [19]. Un défaut peut rapidement dégrader la performance du système ou même endommager le matériel. Il peut aussi exiger des interventions fréquentes des utilisateurs ce qui rend le système moins autonome. En conséquence, une détection rapide préserve le robot et son environnement des risques. Aussi, elle garantit le bon fonctionnement des systèmes engendrant un gain en efficacité et une économie considérable en sauvegardant les matériels et réduisant les coûts et désagréments liés aux interventions humaines. Pour ces raisons, nous nous intéressons à la détection et l'isolement des défauts (Fault Detection and Isolation FDI).

### 2.2 Définitions

### 2.2.1 Défauts

"Un défaut" est défini comme une déviation non autorisée d'au moins une propriété caractéristique d'une variable par rapport à un comportement acceptable [40]. Un défaut peut entraîner *une défaillance*, qui est une altération ou cessation de l'aptitude d'une entité à accomplir une fonction requise [69]. Une entité est en état de panne lorsqu'elle subit une défaillance. Les défauts qui peuvent survenir sur un robot sont variés. Les sources possibles de ces défauts sont les capteurs, les actionneurs, les éléments mécaniques des éléments extérieurs ou les logiciels implémentés. Les risques liés à ces défauts peuvent être d'après [2] et [1] :

- Des risques mécaniques comme le blocage, la casse, ou la dégradation à cause de frottements entre les composants (roues et châssis par exemple).
- Des risques électriques comme une dégradation des fils ou l'explosion des batteries à cause d'un court-circuit sur un composant (moteur ou carte électronique).
- Des risques liés à la navigation comme une navigation vers une position autre que celle prévue à cause d'une perte de position ou d'une imprécision de la carte numérique de l'environnement, d'une défaillance d'un des capteurs donnant la position du robot ou un bug dans l'algorithme de navigation.
- Des risques de collision avec un autre robot, un objet ou un humain, à cause d'un défaut lié au système d'évitement d'obstacles que ce soit dû à un capteur de détection d'obstacles défaillant ou bien une position relative ou absolue pas suffisamment précise.
- Des risques de collision liés à la navigation causés par des sources externes comme : les glissements, la présence de brouillard, des obstacles cachés, des objets mobiles se déplaçant rapidement et donc difficiles à détecter, des petits objets non détectables, des objets non visibles par un laser (objets transparents) ou encore des objets négatifs comme des trous par exemple.
- Des risques liés à des temps de calcul trop importants (surcharge du (des) processeur(s)) rendant les réactions du robot trop tardives.
- Des risques d'échauffement des composants à cause de la température ambiante et d'une durée de fonctionnement importante.

Dans le cadre de cette thèse, les objectifs applicatifs amènent à nous focaliser sur les défauts présentés dans le tableau 2.1.

Type de la source	Source de défaut	
	Système absolu de localisation (GPS)	
	Centrale inertielle (IMU)	
Capteur	Odométrie des roues	
	Système de détection des obstacles (Laser)	
	Communication par UWB	
Externe	Glissement	
Mécanique	Blocage de roues	

Tableau 2.1 – Défauts considérés

En ce qui concerne leurs effets sur le modèle de robot, les défauts peuvent être classés en tant que perturbation additive ou multiplicative du modèle nominal. Par exemple, l'équation (2.1) illustre une représentation d'état d'un système quelconque.  $f_i$  et  $f_o$  sont des paramètres de défauts additifs affectant respectivement l'entrée et la sortie. Les défauts multiplicatifs modifient les fonctions f et h.

$$X(t) = f(t, x(t), u(t)) + f_i$$
  

$$Y(t) = h(t, x(t), u(t)) + f_o$$
(2.1)

Un défaut est aussi caractérisé par son empan temporel. On distingue ainsi un défaut brusque,

d'un défaut naissant (dérive) et d'un défaut intermittent [40]. La figure 2.1 illustre la forme de ces derniers.



FIGURE 2.1 – (a) défaut brusque (b) défaut naissant et (c) défaut intermittent

#### 2.2.2 Diagnostic des défauts

Pour faire face aux différents défauts qui peuvent survenir sur n'importe quel système automatisé, les chercheurs ont travaillé pendant des décennies pour élaborer des méthodes de diagnostic pouvant signaler rapidement ces défauts [32]. La détection et l'isolement de défauts (en anglais Fault Detection and Isolation (FDI)) est le terme utilisé dans la plupart des revues scientifiques pour faire référence au diagnostic de défauts. Cette tâche se compose principalement de deux phases :

- Détection : La détermination de la présence d'un défaut et l'instant de son occurrence, sans forcément avoir une idée sur sa localisation.
- Isolement : En plus de détecter un défaut, l'isolement consiste à déterminer la localisation du défaut et son type.

Deux phases supplémentaires peuvent s'y rajouter :

- Identification : Elle suit la détection et elle sert à déterminer la sévérité et l'empan temporel d'un défaut.
- Accommodation : Cette phase complémentaire consiste à reconfigurer le système pour éviter de perdre l'objectif de commande et pour préserver le robot. Dans certains cas où le défaut est grave et risque d'endommager le matériel, cette phase consiste à faire un arrêt d'urgence.

Le FDI fait l'objet de recherche depuis les années 70. Les chercheurs ont proposé des solutions très diverses pour ce problème. Dans l'espoir d'améliorer l'efficacité et la sûreté des procédés, l'industrie a porté un intérêt croissant à ce domaine, ce qui a contribué à l'enrichissement et à la diversité des solutions proposées.

### 2.3 Problématique

#### 2.3.1 Cahier des charges pour les méthodes de diagnostic

**Concernant les méthodes individuelles de diagnostic** Généralement, l'étude de la FDI est considérée après la conception de la commande dans une phase de supervision. Pour cela, il est souvent impossible d'intervenir sur la conception du contrôleur, par exemple pour ajouter des capteurs redondants. Il est donc nécessaire de concevoir des méthodes de diagnostic se basant sur les informations et les données disponibles. Dans ce cas, les méthodes de diagnostic se basent sur les mesures fournies par les capteurs, les modèles mathématiques connus et les

autres connaissances sur le comportement du système.

L'application d'une méthode de diagnostic sur les robots mobiles demande que certains critères soient respectés :

- Une méthode utilisable en ligne : La majorité des robots mobiles utilisés dans les contextes agricoles et industriels sont autonomes. Le diagnostic doit donc être réalisable alors que le robot réalise la tâche qui lui incombe.
- Une méthode temps réel : La méthode doit être efficace(peu de temps de calcul tout en étant performante) et ne pas dégrader les performances de contrôle du robot (la puissance de calcul disponible sur le robot est généralement limitée).
- Une méthode compatible avec les représentations non linéaires des systèmes : En raison de la nature du robot, les capteurs, les actionneurs et les modèles dynamiques des robots peuvent être non linéaires. La méthode doit pouvoir s'affranchir de cette non-linéarité.

**Concernant le système global de diagnostic d'un robot** Comme nous le montrerons dans le paragraphe 2.6, une seule méthode de diagnostic n'est pas suffisante pour traiter la totalité des défauts considérés, d'où la nécessité d'un système global de diagnostic. Il est composé de plusieurs méthodes de diagnostic mises en parallèle ou configurées d'une façon plus complexe. Ce système global doit être suffisamment performant pour satisfaire les critères suivants :

- Un diagnostic rapide : Pour empêcher le défaut d'engendrer d'importants dégâts , il s'avère important de détecter puis d'isoler un défaut le plus rapidement possible voire de l'anticiper.
- Un diagnostic précis : Le fait d'interrompre fréquemment les opérations suite aux fausses alarmes peut réduire drastiquement la productivité des robots. Pour cela, il est primordial de minimiser les fausses alarmes. Toutefois, en même temps, les défauts ne doivent pas passer inaperçus pour le système.
- Un diagnostic peu coûteux : Afin de ne pas dégrader la mission du robot, la tâche de diagnostic doit être la moins coûteuse possible en terme de temps de calcul et de volume de stockage. Elle doit en outre utiliser le minimum de capteurs supplémentaires possibles.
- Un diagnostic adaptable : Le système de diagnostic doit être facilement adaptable. L'application à des robots ou dans des contextes différents doit être possible avec un nombre de modifications minimal.
- Un diagnostic réactif et révélateur : Nous proposons ici une supervision du robot faite par un opérateur à distance (poste de supervision). Suite à l'isolement d'un défaut, le système de diagnostic avertit l'opérateur d'une manière claire et efficace si son intervention est requise.

### 2.3.2 Difficultés

**Changement de paramètres du système selon le contexte** Le respect de ce cahier des charges n'est pas évident pour plusieurs raisons. D'abord, un robot mobile dans des contextes agricoles ou industriels peut se retrouver dans différents environnements et peut subir des changements de certains de ces paramètres. Par exemple, un robot roulant dans un champ rencontre différentes conditions d'adhérence ou encore le poids total d'un robot transportant des objets dans une usine, change selon le poids de l'objet transporté.

**Domaine des méthodes et diversité des défauts** Ensuite, d'une part les sources possibles de défauts et leurs caractéristiques sont nombreuses tel qu'expliqué dans la section 2.2.1. D'autre part, chaque méthode de diagnostic possède des limites et des conditions d'application spécifiques [11]. En conséquence, la conception d'une méthode globale pouvant diagnostiquer une liste de défauts est une opération compliquée.

Limite de l'intelligence embarquée Même si les méthodes de diagnostic sont employées de façon optimale, elles peuvent se tromper de temps en temps. En effet, la littérature montre qu'une méthode n'est jamais fiable à 100% [36,40,53,54]. Ce qui prouve que l'intelligence embarquée est toujours limitée en performance. Alors, il faut pouvoir autoévaluer continuellement le dispositif.

### 2.3.3 Hypothèses proposées

Dans cette section, nous présentons les hypothèses proposées pour pallier à ces difficultés.

D'abord, en ce qui concerne le changement des paramètres du robot, la plupart des méthodes de diagnostic possèdent des paramètres qui peuvent être ajustés en temps réel, comme par exemple, les seuils de détection ou les matrices caractéristiques d'un filtre de Kalman [44]. Ainsi, en adaptant certains paramètres, il sera possible d'adapter en ligne ces méthodes au changement du contexte.

Ensuite, en sachant que chaque méthode de diagnostic possède ses propres limites, et vue la diversité des défauts, l'idéal sera d'adopter une combinaison optimale de méthodes afin de profiter des points forts de chacune. À titre d'exemple, imaginons que pour diagnostiquer un défaut  $D_1$  dans un contexte  $C_1$ , une méthode  $M_1$  est plus performante que  $M_2$ . Par contre,  $M_2$  est meilleure dans le contexte  $C_2$ . Dans ce cas, pour mieux diagnostiquer  $D_1$ , le choix de la méthode à appliquer va dépendre du contexte dans lequel se trouve le robot actuellement.  $M_1$  doit être appliquée dans  $C_1$ , et  $M_2$  dans  $C_2$ .

Finalement, pour compenser les limites des algorithmes, l'intervention de l'humain pourra être de grande valeur. Grâce à ses capacités cognitives et d'adaptation, une personne peut jouer plusieurs rôles importants dans la boucle de diagnostic :

- Assister à l'isolement suite à la détection d'un défaut ;
- Assister au basculement entre les méthodes selon les contextes;
- Évaluer en temps réel le comportement de la méthode globale de diagnostic;
- Assister à la décision prise a posteriori suite au diagnostic d'un défaut ;
- Et bien sûr, un humain réparera les éléments défectueux.

Ainsi, dans cette thèse, nous proposons une méthodologie permettant de concevoir un système global pouvant diagnostiquer une liste de défauts de la robotique mobile en temps réel. Il est composé de deux blocs principaux :

- Un système de Supervision du Diagnostic de Défauts Robot (S2D2R) : Il contient deux blocs : la Méthode Hybride de Diagnostic (MHD) formée de plusieurs sous-méthodes de diagnostic et un Module d'Interaction Homme/Robot (MIHR) chargé de faire le lien entre la MHD et l'humain tout en aidant l'humain à proposer des solutions pertinentes pour chaque situation.
- Un bloc d'outils de Modélisation, de Paramétrage et d'Adaptation des Méthodes de Diagnostic aux Défauts (OMOPAMED).

Les méthodes traitant chaque défaut seront évaluées afin d'identifier la ou les meilleures. Ensuite, leurs paramètres seront ajustés automatiquement en temps réel dans le but d'optimiser leurs performances grâce à l'évaluation continue par l'opérateur humain des décisions prises par la MHD. Enfin, suite au diagnostic d'un défaut, la MIHR communiquera des informations suffisantes et efficaces à l'opérateur en lui donnant un rôle selon l'amplitude du défaut, son type et son niveau de diagnostic (c.à.d détection ou isolement).

### 2.3.4 Méthodologie de recherche

Dans le but de concevoir la MHD et le S2D2R, suite à l'analyse du contexte applicatif de cette thèse, une liste de défauts considérés est identifiée, puis une méthodologie scientifique est respectée. Ses étapes principales sont comme suit :

- a) Études des caractéristiques des défauts considérés :
  - i Chaque défaut est étudié en déterminant ses caractéristiques comme sa source, son empan temporel, son modèle et ses effets sur le comportement de robot.
  - ii Connaissant les caractéristiques d'un défaut, on identifie les types de méthodes les mieux adaptés pour le diagnostic.
- b) État de l'art des méthodes de diagnostic : Suite à la détermination des caractéristiques des défauts, une recherche bibliographique est faite portant sur les différentes méthodes de diagnostic capables de détecter et/ou isoler chacun des défauts. À la fin de cette étude, une ou plusieurs méthodes de diagnostic sont attribuées à chaque défaut.
- c) Élaboration de la forme initiale de la MHD :
  - i Ajustement et optimisation des méthodes : Les méthodes de diagnostic sont adaptées et paramétrées afin d'avoir une performance optimale.
  - ii Évaluation et intégration dans la MHD : Suite à l'optimisation, les méthodes sont évaluées puis comparées entre elles. La ou les méthodes ayant la meilleure performance seront intégrées dans la MHD. La combinaison des méthodes n'est pas forcément simple, des méthodes peuvent en alimenter d'autres par leurs résultats pour une meilleure précision.
- d) Définition du mode d'interaction :
  - i Définition de la base de connaissances initiale : Cette base définit les règles du comportement de la méthode dans chaque situation (détection ou isolement d'un défaut).
  - ii Communication en cas de défaut : Un protocole de communication est défini afin d'optimiser la transmission de l'information à l'opérateur pour qu'il réagisse de la manière la plus efficace possible.
  - iii Évaluation et adaptation des méthodes : L'opérateur évaluera les décisions prises par la MHD. Grâce à cette évaluation, la MHD subira des modifications soit par l'adaptation des paramètres des méthodes composant la MHD, soit par la modification de sa forme initiale i.e. l'application d'autres sous méthodes au lieu de celles initialement choisies.
  - iv Mise à jour de la base de connaissances : La pertinence de la décision prise suite à chaque diagnostic est évaluée par l'opérateur. Selon cette pertinence, la base initiale de connaissances est modifiée pour mieux réagir dans le futur.
- e) Validation des propositions : Les hypothèses proposées sont appliquées sur un cas réel pour les valider.

Les paragraphes suivants de ce chapitre présentent les méthodes de diagnostic des défauts utilisées dans la littérature et leur adéquation avec les défauts considérés.

## 2.4 Généralités sur les méthodes de diagnostic

Nous avons constaté dans 2.2.1 que les défauts diffèrent selon leurs sources, leurs natures (additif ou multiplicatif) et leurs caractéristiques temporelles. Il en est de même pour les méthodes de diagnostic pour lesquelles il existe une grande variété. D'une manière générale, selon leur mode de conception et d'application, ces méthodes sont réparties en quatre catégories principales [81].

- 1. Les méthodes à base de modèle : En se basant sur un modèle mathématique du système et parfois du défaut, ces méthodes génèrent souvent des valeurs appelées résidus. Le diagnostic se fait généralement ensuite par analyse de ces résidus.
- 2. Les méthodes à base de connaissances : Ces méthodes prédisent l'état du système à partir des valeurs de ses entrées/sorties et des connaissances liées au comportement du système dans les cas de fonctionnement normaux et en cas de défauts.
- 3. Les méthodes à base de données : Ces méthodes "voient" le système comme une boîte noire. Avec une connaissance limitée du modèle du système et de celui des défauts. Elles diagnostiquent les défauts en analysant les entrées et les sorties du modèle.
- 4. La redondance matérielle : L'idée ici est de dupliquer le capteur ou d'ajouter d'autres capteurs permettant de mesurer la même variable.

Les méthodes à base de modèle Les méthodes à base de modèle ont été largement étudiées depuis les années 1970 [83]. Dans cette approche, les défauts sont classés généralement en deux catégories principales : additifs et multiplicatifs. La structure générale de ces méthodes est illustrée à la figure 2.2.



FIGURE 2.2 – Structure générale des méthodes à base de modèle

Les techniques à base de modèle sont applicables lorsque le système est modélisable. Dans le cas d'un modèle exact relativement simple, elles peuvent être très précises et très efficaces (nécessité de peu de puissance de calcul). Cependant, et notamment dans les cas des systèmes à certains degrés de non linéarité, elles peuvent devenir moins précises, plus coûteuses et plus complexes à concevoir.

Selon le mode de génération des résidus, les méthodes à base de modèle se répartissent en trois catégories : les méthodes basées sur les observateurs [14], les méthodes basées sur l'espace de parité ou redondance analytique [55] et les méthodes d'estimation de paramètres [60]. Il est important de noter aussi qu'il existe des méthodes graphiques à base de modèle comme indiqué dans [62]. En raison de la capacité des méthodes graphiques à étudier les propriétés de diagnostic hors ligne, elles ont été proposées comme alternative aux modèles entièrement analytiques et numériques.

Les observateurs sont des outils connus et reconnus dans le domaine de la théorie du contrôle. Ils estiment itérativement l'état complet ou partiel d'un système. Les observateurs sont utilisés dans de nombreuses applications robotiques comme pour le remplacement de capteurs physiques, la prédiction des états internes, le contrôle robuste et le diagnostic [50]. Les résidus sont obtenus par la comparaison entre la valeur estimée et la valeur correspondante mesurée grâce aux capteurs. L'utilisation d'un observateur pour le FDI n'exige pas la connaissance du modèle d'un défaut pour le détecter. Mais la conception des observateurs n'est pas toujours possible ou évidente.

Ensuite, le diagnostic basé sur l'analyse de relations de parité est une méthode naturelle de diagnostic qui consiste à vérifier la consistance entre les différentes variables d'un système. Elle est basée sur la construction de variables à partir de deux sources a minima (modèles, capteurs, entrées et sorties). Une comparaison de celles-ci conduit à la génération de résidus. Cette approche nécessite peu de puissance de calcul. Cependant, un post-traitement est obligatoire pour obtenir des résultats satisfaisants. Le seuillage simple est sensible au bruit.

Enfin, nous pouvons estimer des variables représentant la contribution du défaut au modèle du robot à l'aide des algorithmes d'estimation de paramètres. Ces paramètres peuvent être ensuite utilisés pour le paramétrage de certaines méthodes de diagnostic comme, par exemple, l'estimation récursive des paramètres des défauts [60] ou encore l'estimation multi-modèles [10]. Une connaissance préalable des défauts et leur influence sur le modèle est nécessaire pour pouvoir appliquer ce type de méthode. La précision de ces méthodes est tributaire de la précision des modèles admis de défauts.

La littérature est riche de contributions à base de modèle pour le diagnostic de défauts. Ces méthodes ont été appliquées pour la surveillance à bord de la dynamique de véhicules ferroviaires par exemple [71]. Grâce à l'approche multi-modèle, les défauts capteurs et actionneurs sur ces machines sont diagnostiqués. Le problème de faible adhérence a été géré par l'estimation à l'aide de filtres de Kalman. Le défaut sur la suspension est détecté en combinant les méthodes à base de modèle et les méthodes à base de données. Selon la revue [83], les différentes techniques de FDI à base de modèle ont été appliquées pour surveiller les éoliennes, les véhicules, les moteurs, les processus chimiques et les batteries.

Les méthodes à base de connaissances Les méthodes basées sur la connaissance sont basées sur la connaissance heuristique de la procédure (i.e. le système à diagnostiquer). Dans cette approche, comme il est illustré à la figure 2.3, un modèle de système analytique n'est pas nécessaire, car la méthode adopte uniquement les informations extraites des sorties du système. De cette manière, seule une connaissance empirique du système est utilisée et le diagnostic se fait de manière heuristique ce qui pourrait être avantageux dans le cas de systèmes difficilement modélisables. Les conditions d'application de ces approches sont généralement moins lourdes que celles des méthodes à base de modèle. Par contre, ces méthodes sont généralement coûteuses en



FIGURE 2.3 – Structure générale des méthodes à base de connaissances

termes de temps et de puissance de calcul.

Dans [82], des techniques basées sur les connaissances sont présentées. Des procédures de logique floue y sont adoptées pour cartographier les entrées et les sorties des systèmes dynamiques. Des filtres à particules ont été appliqués pour le diagnostic de situations d'un rover spatial [23] et d'un robot serveur [22]. Un filtre à particules à abstraction mixte (ou à résolution variable, i.e. qui peut regrouper plusieurs modes similaires dans un seul super mode) est adopté dans [63] pour l'estimation d'état en robotique mobile, y compris la détection de deux défauts. L'applicabilité de plusieurs variantes de cet algorithme a été prouvée sur les systèmes non linéaires [38].

Les approches à base de connaissances peuvent être combinées avec des approches à base de modèle comme le filtre à particules gaussien utilisé dans [73] pour diagnostiquer des défauts moteurs dans un robot autonome sous-marin.

Les méthodes à base de données En se basant sur une base de données plus ou moins importante, les méthodes à base de données peuvent diagnostiquer l'état d'un système en le traitant généralement comme une boîte noire. Les méthodes rentrant dans cette catégorie sont très nombreuses. Les techniques principales sont illustrées à la figure 2.4. Ces méthodes peuvent être combinées avec des approches à base de connaissances ou à base de modèle. Elles peuvent même être appliquées sur les sorties des autres méthodes puisqu'elles ne dépendent pas des données en entrée. Toutefois, ces méthodes souffrent d'un inconvénient principal : la quantité de données et le temps de calculs nécessaires pour l'entraînement.

Dans la littérature, ces méthodes ont été appliquées dans de nombreux domaines. En effet, les approches basées sur les statistiques, sur l'intelligence artificielle et sur le traitement de signal ont eu un succès dans les systèmes de traitement chimique [57]. Des chercheurs les ont appliquées pour le diagnostic des défauts de matériels comme les vannes, de capteur comme ceux de température, des actionneurs, et d'autres changements non souhaités de paramètres. Ensuite, afin de surveiller l'état de santé des éoliennes, les réseaux de neurones et les méthodes à base de la logique floue ont fait leur apparition pour le diagnostic de différentes étapes de la génération d'électricité [35]. Selon la revue [61], les méthodes étaient capables de diagnostiquer des défauts dans les stations d'épuration des eaux usées. Grâce à ces approches, le diagnostic



FIGURE 2.4 – Les méthodes de FDI à basse de données

de l'état de divers capteurs et matériels est assuré. Finalement, les auteurs de [8] ont recensé l'utilisation de différentes méthodes à base de données pour le diagnostic de processus. Selon cette investigation, à la date de publication de la revue (mars 2018), plus de 4000 documents scientifiques ont été publiés ce qui prouve le succès que les méthodes à base de données ont rencontré dans ce domaine.

La redondance matérielle L'idée la plus basique pour surveiller un capteur est la duplication de ce capteur ou l'ajout d'autres capteurs pour mesurer la même variable. Cette méthode s'appelle la redondance matérielle. Pour utiliser cette méthode, le système doit pouvoir supporter des capteurs redondants. Son architecture est décrite à la figure 2.5.



FIGURE 2.5 – La redondance matérielle

Dans [55], les deux approches, relations de parité et redondance matérielle, sont mises en œuvre pour le diagnostic de défauts capteur sur un robot à 4 roues équipés de 11 capteurs (capteurs de vitesse, d'orientation et de localisation) regroupés en 5 groupes (les capteurs d'un même groupe mesure les mêmes variables). Les relations de parité obtenues entre ces capteurs ont fournis 18 résidus. Les défauts à détecter et isoler sont le glissement, la présence d'un bruit important sur un capteur d'orientation, une surtension sur un capteur et tout biais de mesure. L'évaluation des méthodes a été faite en simulation. Même si les taux de performance de détection et d'isolement de ces deux méthodes ne sont pas précisés, les résultats décrits montrent une détection fiable et instantanée de l'ensemble des défauts injectés lors de la simulation. En pratique, cette méthode est essentiellement utilisée pour détecter et isoler les défauts capteurs.

Malgré sa simplicité, la redondance matérielle est une méthode très limitée de par le fait qu'elle est très coûteuse et peut rapidement surcharger le système.

# 2.5 Classification des méthodes répondant au cahier des charges

Nous avons vu qu'il existe une large variété de méthodes de diagnostic. Certaines d'entre elles ne sont pas compatibles avec les critères définis dans la partie 2.3.1. Citons, par exemple, les méthodes de traitement du signal. Qu'elles soient temporelles ou fréquentielles (transformation de Fourrier ou ondelette), ces méthodes sont utilisées pour diagnostiquer notamment les machines tournantes. Elles sont gourmandes en puissance de calcul et certaines de ces méthodes ne peuvent fonctionner que hors ligne [54].

D'autres répondent aux critères listés dans le cahier des charges. Le Tableau 2.2 détaille les principales méthodes de diagnostic utilisées ou susceptibles d'être utilisées en diagnostic de robots mobiles.

Il est important de préciser que ce tableau ne représente pas une liste exhaustive des méthodes pouvant être appliquées en diagnostic de robots mobiles. Il est issu d'une recherche bibliographique visant les techniques de diagnostic appliquées avec succès dans ce domaine et applicable à notre cas d'étude.

Catégorie	Méthodes	
	Filtres de Kalman : standard, étendu et unscented	
À hann de mardèlar	Estimation récursive de paramètres	
A base de modeles	Estimation multi-modèle	
	Relation de parité (redondance analytique)	
À base de connaissance	Filtre à particules standard et ses variantes	
À base de données	Réseaux de neurones notamment l'apprentissage profond	

Tableau	2.2 -	$\operatorname{Les}$	méthodes	considérées
---------	-------	----------------------	----------	-------------

## 2.6 Attribution des méthodes de diagnostic aux défauts

Dans ce paragraphe, nous montrons un extrait des contributions faites dans le domaine du diagnostic en robotique mobile pour pouvoir attribuer les méthodes de diagnostic adéquates pour chaque défaut.

**Méthodes à base de modèle** D'abord, les méthodes à base de modèles ont été largement appliquées pour le diagnostic des robots mobiles. Étant réservé aux modèles linéaires, le Filtre de Kalman (KF) standard ne promet pas des résultats acceptables [44]. En revanche, ses deux variantes, le Filtre de Kalman Étendu (EKF) et unscented (UKF) sont un succès.

Une technique FDI est décrite dans [64] où quatre défauts sont détectés et isolés grâce à un EKF sur un robot mobile de type skid-steering à deux roues. Suite à la détection par l'EKF, une comparaison par paire entre les sorties des modules de positionnement produit des résidus qui permettent de définir une table de vérité. Ce tableau permet d'isoler les états : normal, défaut encodeur, défaut du moteur à courant continu sans balai, défaut laser et défaut du système de navigation laser.

Dans le but d'améliorer la fusion de données et de gagner en précision, une variante du KF, à savoir le filtre informationnel (IF), a été adopté dans [7] où les mesures erronées ont été identifiées grâce aux résidus calculés à l'aide de l'IF et associés à chaque mesure. Ce résidu est la divergence globale de Kullback-Leibler qui mesure la divergence entre la distribution de probabilité prédite par l'IF à l'instant t - 1 et celle mesurée à l'instant t. La mesure est déclarée comme erronée si la valeur du résidu dépasse un seuil prédéfini. Dans la même optique, une contribution annexe visant à définir un seuil adaptatif pour les méthodes qui détectent les défauts par seuillage est présentée dans [21].

Tout comme l'EKF et l'KF, l'UKF peut être utilisé pour les problèmes de détection et d'isolement. La supériorité de l'UKF sur l'EKF dans l'estimation de systèmes hautement non linéaires est démontrée dans [9] où les deux algorithmes sont testés sur un pendule inversé. En revanche, dans une comparaison entre un EKF et un UKF pour la tâche de localisation d'un robot équipé de cinq capteurs à ultrasons [27], les résultats étaient proches (UKF légèrement supérieur). En effet, le système n'est pas hautement non linéaire et le coût de calcul de ce processus est faible. Ces deux algorithmes peuvent être intégrés dans une architecture d'estimation multi-modèle (MMAE) en appliquant plusieurs filtres dont chacun représente un mode spécifique. Une MMAE à base d'EKF a été adoptée sur un robot mobile pour la FDI des défauts roues, encodeurs et gyroscope dans [58]. Testée en simulation, cette technique a permis d'isoler de manière efficace les défauts considérés. Une méthode semblable est présentée dans [29] où les filtres utilisés dans l'architecture sont des UKF.

Dans [55], les relations de parité ont permis de diagnostiquer dix défauts capteurs sur un véhicule autonome à quatre roues. Les relations entre les mesures redondantes de différents capteurs et le modèle de déplacement du robot aboutissent à la génération de résidus. Le seuillage de ces résidus permet d'isoler le défaut et intervient pour la reconfiguration du système pour compenser le défaut. De plus, dans [47], en se basant sur la redondance analytique, les auteurs ont défini des résidus structurés permettant la FDI de plusieurs défauts de type actionneur bloqué et capteur subissant des dérives.

En ce qui concerne l'estimation récursive de paramètres, cette technique a été utilisée dans [45] dans le but d'assurer une commande tolérante aux défauts. La présence de défauts apparaît comme une déviation dans les paramètres estimés.

En bref, en fonction de la complexité du modèle, une méthode à base de modèle peut être très précise et très efficace (nécessité de peu de puissance de calcul) dans le cas d'un modèle relativement exact et simple. Elle peut cependant devenir moins précise, plus coûteuse et plus complexe à concevoir dans certains cas. Toutes les versions de KF n'exigent pas de modèle de défaut pour détecter un défaut. C'est une méthode parfaite pour détecter les défauts de type perturbation. Cependant, un bruit plus ou moins proche de Gauss est nécessaire pour maintenir la précision. Le KF fonctionne parfaitement sur les modèles linéaires, l'EKF fonctionne bien sur les modèles non linéaires de premier ordre et l'UKF sur les modèles de second ordre.

La MMAE peut être considérée comme une méthode à la fois à base de modèle et à base de connaissances puisqu'une connaissance sur le défaut est exigée. Elle peut isoler de nombreux défauts, mais nécessite un modèle en cours d'exécution pour chacun d'eux. Une estimation de paramètres a priori peut être effectuée pour estimer les paramètres des défauts. Elle peut devenir très coûteuse pour plusieurs défauts et est principalement utilisée pour détecter des défauts additifs et multiplicatifs connaissant leur forme. L'estimation récursive de paramètres partage les mêmes caractéristiques que la MMAE, mais elle est plus sensible au bruit.

La génération de résidus par la redondance analytique nécessite peu de puissance de calcul généralement. Cependant, un post-traitement est obligatoire pour obtenir des résultats satisfaisants. Le seuillage simple est sensible au bruit.

Les méthodes à base de connaissances Pour diagnostiquer le robot, ces méthodes s'appuient sur des connaissances sur le comportement du robot ou de certains paramètres dans chacun des modes possibles pour le diagnostiquer. Le filtre à particules (PF) et ses variantes rentrent dans cette catégorie. Le PF standard a été appliqué dans [26] pour assurer une localisation robuste malgré la présence éventuelle de défauts. Pour diagnostiquer l'état du rover de Mars, différentes variantes de PF on été appliquées dans plusieurs contributions : le PF à résolution variable dans [78], la version Rao-Blackwell intégrant un UKF dans [22] et le PF gaussien dans [77]. Les auteurs ont essayé de diagnostiquer l'état de ce robot hautement autonome qui devait évoluer très longtemps sur le sol de la planète rouge. À l'aide de leurs algorithmes, ils ont distingué des situations comme roues bloquées par des roches ou des actionneurs en panne. Dans [76], un filtre à particules a été adopté pour le diagnostic de défauts de l'asservissement visuel basé sur l'image (vision par caméra). Dans cette étude, un filtre à particules possédant 500 particules s'est montré plus performant qu'un UKF (erreur quadratique moyenne égale à 0.04 pour le PF contre 0.089 pour l'UKF). Par contre, le temps de calcul consommé par le PF (19.10<sup>-3</sup>s) est largement supérieur à celui de l'UKF (7.10<sup>-3</sup>s).

Les méthodes à base de connaissances dépendent moins du modèle, mais une bonne connaissance du comportement du système dans chaque état (état défaillant, état normal) est importante. Ces méthodes sont généralement plus coûteuses en calcul que celles basées sur un modèle. Toutes les versions de PF considèrent le système comme un système hybride, c.à.d que le système est constitué de variables continues (vitesses, angles) et de modes discrets (normal, défaut).

PF est très général, aucune hypothèse n'est formulée. Cependant, il demande une très grande puissance de calcul pour détecter les défauts les moins probables. Le PF standard doit être évité alors dans notre cas. Sa version Rao-Blackwell est une bonne alternative, mais les variables continues doivent suivre un modèle linéaire, car elle utilise un KF pour l'estimer. Enfin, et en faisant l'hypothèse d'une puissance de calcul importante, la version gaussienne est très puissante dans le cas de modèles non linéaires.

Les méthodes à base de données Les méthodes qui rentrent dans cette catégorie sont très nombreuses comme expliqué dans la partie 2.5. Nous nous attardons ici uniquement sur les méthodes basées sur l'intelligence artificielle.

Des défauts capteurs d'un robot mobile à six roues ont été isolés dans [33] grâce à un algorithme

d'apprentissage profond (Deep Learning (DL) en anglais) de type réseau de croyance profonde (Deep Belief Network DBN). Les réseaux de neurones sont appliqués dans [25], où les algorithmes classiques comme le feed-forward multi-couches ont réussi à avoir un taux de prédiction de défaut supérieur à 95%. Les auteurs de [26] ont appliqué un réseau de neurones sur la sortie d'un filtre à particules pour le diagnostic de robots mobiles dans des environnement inconnus. Leur méthode isole des défauts connus et détecte la présence de défauts inconnus. Pour illustrer un exemple où l'intelligence artificielle est appliquée pour traiter la sortie d'une méthode à base de modèle, nous citons [42] où les auteurs combinent l'EKF, l'estimation de paramètres et le réseau de neurones pour isoler les défauts capteurs et actionneurs sur un drone quadrotor. Pour prédire l'erreur sur l'estimation de la pose du robot utilisant un algorithme de localisation se basant sur le PF Rao-Blackwell, un algorithme d'apprentissage profond, à savoir, le réseau de neurones convolutif (CNN), a été utilisé dans [5]. La prédiction d'erreur a contribué à l'amélioration de la robustesse en réduisant l'effet des bruits de mesure.

En bref, les méthodes basées sur les données sont très intéressantes. Les techniques les plus utilisées semblent être celles de l'apprentissage profond. Leur application nécessite une grande base de données pour l'entraînement, mais les résultats sont généralement satisfaisants. En raison de sa généralité, il peut être utilisé de plusieurs manières. Par exemple, il peut être utilisé directement sur la mesure et la commande ou pour traiter les résidus d'approches basées sur un modèle.

Finalement, les tableaux 2.3, 2.4 et 2.5 récapitulent les conditions d'applications, les avantages et les inconvénients des méthodes considérées dans la littérature. Grâce à ces tableaux, il est possible d'attribuer les méthodes de diagnostic aux types de défauts. Par exemple, pour les défauts de bruits sur les capteurs de localisation possédant un modèle connu,l'EKF ou l'UKF sera un bon choix. Les défauts modélisables sont isolables à l'aide de la MMAE. L'intelligence artificielle est réservée aux cas où la construction d'une base de données représentant les différents modes est possible. Toutefois, il est clair que chaque méthode possède des limites qui l'empêchent d'être appliquée pour tous les défauts à la fois.

À la lumière de ces tableaux et en considérant les défauts sur le tableau 2.1, nous constatons que les méthodes basées sur l'intelligence artificielle sont envisageables pour traiter tous les défauts à condition d'avoir suffisamment de données pour l'entraînement. Les méthodes à base de modèle sont intéressantes pour les défauts de type de perturbation (capteurs).

Méthode	Conditions d'application	Inconvénient	Avantage
Réseau de Nécessité d'avoir suffisam-		Dépendant des données	Approche très générique
neurones ment de données pour l'ap-		d'apprentissage et de	
convention-	prentissage	la conception du réseau	
nel		de neurones. De plus,	
		déboguage très difficile	
DL	Base de données pour l'en-	Taille de la base de	Approche générique qui a
	traînement plus importante	données et temps	prouvé ses performances
que celle de réseau de neu-		d'entraînement. Et,	dans plusieurs domaines
rones conventionne		déboguage très difficile	

Tableau 2.3 – Caractéristiques des méthodes à base de données

Méthode	Conditions d'application	Inconvénient	Avantage
KF	Système linéaire. Détecte	Dégradation en perfor-	Estimateur optimal pour
	bien les défauts de type per-	mance si le bruit est non	les systèmes linéaires avec
	turbation	gaussien	bruits gaussiens
EKF	Système non linéaire jus-	Calcul de la matrice ja-	Haute performance pour un
	qu'à l'ordre 2. Détecte bien	cobéenne coûteux en temps	système de 1er ordre de non-
	les défauts de type pertur-	de calcul pour certains	linéarité avec distributions
	bation	systèmes	gaussiennes
UKF	Système non linéaire jus-	Temps de calcul plus impor-	Converge plus vite qu'un
	qu'à l'ordre 3. Détecte bien	tant que pour un EKF pour	EKF pour les systèmes non
	les défauts de type pertur-	les systèmes à faible ordre	linéaires
	bation	de non linéarité	
Relations	Possibilité de calculer la	Peu robuste au bruit	Temps de calcul faible
de parité	même variable avec plu-		
	sieurs méthodes		
Estimation	Architecture du défaut	Peu robuste au bruit	Adaptée aux défauts multi-
récursive	connue		plicatifs et additifs
de pa-			
ramètres			
Estimation	Architecture connue du	Performance dépendante du	Adaptée aux défauts multi-
multi-	défaut	post-traitement des résidus	plicatifs et additifs
modèle			

Tableau 2.4 – Caractéristiques des méthodes à base de modèle

Méthode	Conditions d'application	Inconvénient	Avantage
PF stan-	Connaissance a priori des	Un grand nombre de parti-	Théoriquement, aucune ap-
dard	caractéristiques des défauts	cules ( et temps de calcul)	proximation réalisée sur le
		est nécessaire pour détecter	système et il fonctionne sur
		les défauts à faible probabi-	n'importe quel modèle
		lité d'occurrence	
PF sen-	Nombre limité de défauts	Dégradation des perfor-	Détecte les défauts peu pro-
sible aux	pouvant être surveillés	mances dans le cas de	bables mieux que PF
risques		nombreux états de défaut	
		similaires	
PF à	Présence d'états semblables	Méthode non générique, car	Temps de calcul faible et
résolution		pertinente uniquement en	haute performance en cas
variable		cas de présence d'états sem-	de modèles de transition et
		blables	d'observation similaires
PF Rao-	Variables continues : condi-	Les mêmes inconvénients de	Des performances
Blackwell	tions de KF	KF pour l'estimation des	supérieures à PF avec
		variables continues	le même nombre de parti-
			cules
PF gaus-	Variables continues : condi-	Les mêmes inconvénients	Des performances
sien	tions d'UKF	d'UKF pour l'estimation	supérieures à PF avec
		des variables continues	le même nombre de parti-
			cules

Finalement, les méthodes à base de connaissances peuvent diagnostiquer les défauts de blocage de roues, puisque le robot aura un comportement prévisible (effet modélisable) lorsqu'il subit un blocage d'une de ses roues.

## 2.7 L'importance de l'humain

Comme nous l'avons évoqué dans la partie 2.3.3, l'humain jouera un rôle important et interagira avec le robot de plusieurs manières. L'humain peut coopérer avec le robot pour réussir à réaliser une mission finale commune. Mais, il peut aussi donner un coup de main au robot en cas de difficulté. Pour cela, les chercheurs s'intéressent de plus en plus à l'étude de l'interaction hommerobot.

Les auteurs dans [15] ont défini un système d'interaction pour assurer une coopération optimale entre un groupe de robots industriels et leur coéquipier humain afin de coopérer d'une manière plus fluide.

Des recherches portant sur le problème de diagnostic de robots mobiles ont été faites aussi. Les auteurs de [30] propose un schéma de contrôle tolérant aux défauts où l'opérateur réagira selon la sévérité du défaut. Il reconfigure des paramètres, choisit un mode dégradé, prend en main le contrôle ou arrête le robot. Dans [17], un système d'aide au diagnostic a été conçu dans le but d'aider l'opérateur à mieux isoler un défaut. Finalement, dans [74], une méthodologie est proposée. Cette méthodologie permet d'évaluer le niveau d'importance des informations pour décider quelle information envoyer à l'opérateur diagnostiquant le système.

## 2.8 Les contributions de la thèse

Nous résumons les contributions majeures de cette thèse dans ce paragraphe.

Un Système de Supervision du Diagnostic de Défauts Robot (S2D2R) est proposé. Il est composé de deux blocs : une Méthode Hybride de Diagnostic (MHD) englobant plusieurs techniques de diagnostic et un Module d'Interaction Homme/Robot (MIHR) permettant une interaction intelligence avec l'humain. Une forme de la MHD est élaboré ensuite en considérant un cas d'application.

Un indicateur de performance permettant l'évaluation objective des méthodes de diagnostic est proposé. En se basant sur cet indicateur, les méthodes éligibles pour intégrer la MHD sont choisies.

Plusieurs contributions sont faites au niveau des méthodes de diagnostic pour la détection et l'isolement des divers défauts. Les défauts capteurs de localisation sont isolés à l'aide d'une technique intégrant deux Filtres de Kalman (KF). Puisque la performance d'un KF est tributaire de son paramétrage, un algorithme basé sur l'algorithme génétique est conçu. Il permet le paramétrage automatique d'un KF dans le but d'optimiser ses performances en détection de défauts.

Une méthode pour la détection des perturbations du système d'évitement d'obstacles (laser) est proposée. Elle se base sur le KF.

Une démarche pour le diagnostic des défauts à modèle connu (additif ou multiplicatif) est définie. Après la modélisation des défauts additifs et multiplicatifs, les modèles obtenus seront intégrés dans des méthodes à base de connaissances. Dans ce but, trois méthodes sont proposées : un
schéma d'estimation multi-modèles intégrant l'UKF et le filtrage passe-bas, un algorithme d'estimation récursive de paramètres intégrant un algorithme de moindres carrés récursifs et l'analyse discriminante linéaire et une version mixte du filtre à particules combinant ses deux variantes standard et gaussienne pour le diagnostic de défauts connus.

Des résidus sont générés à l'aide des relations de redondance. Des défauts capteurs et mécaniques sur un robot mobile sont isolés grâce au seuillage de ces résidus. Ensuite, deux modèles d'apprentissage profond sont proposés pour traiter ces résidus afin de mieux isoler les défauts.

Finalement, une méthode permettant l'isolement des tags défectueux pour la communication UWB est proposée.

# Chapitre 3

# Méthodologie de conception d'un système homme/robot de diagnostic de robot mobile

# 3.1 Schéma global de la méthodologie



FIGURE 3.1 – Forme globale

Dans cette section, nous expliquons la démarche permettant la conception du Système de Supervision du Diagnostic de Défauts Robots (S2D2R) et son fonctionnement.

Comme conclu dans le paragraphe 2.6, il est impossible de traiter tous les défauts considérés à l'aide d'une seule méthode de diagnostic. Le S2D2R est composé de deux parties : 1- la Méthode Hybride de Diagnostic (MHD) qui est constituée de plusieurs sous méthodes de diagnostic organisées de telle façon à ce qu'on puisse tirer le maximum de bénéfices de chacune et, 2- un Module d'Interaction Homme/Robot (MIHR) qui est en charge d'assurer une interaction intelligente avec l'opérateur humain. La forme globale de la démarche permettant la conception de ce système est illustrée à la figure 3.1. La méthodologie est composé de deux parties : une partie représentant les Outils de Paramétrage et d'Adaptation des MÉthodes de Diagnostic aux Défauts (OMOPAMED2) est mise en œuvre hors ligne et en amont de la supervision du robot, et le S2D2R qui fonctionne en ligne. Le premier sous-système (OMOPAMED2) représente la démarche aboutissant à l'élaboration de la MHD, il est composé des éléments suivants :

- Identification des défauts : en se basant sur la configuration du robot ainsi que sur les besoins de l'application, les défauts potentiels sont identifiés.
- Sélection des méthodes de diagnostic : comme décrit dans le chapitre précédent, les méthodes de diagnostic adéquates sont associées à chaque type de défaut en se basant sur la littérature. Cette phase permet d'identifier les méthodes applicables pour chaque défaut.
- Adaptation des méthodes sélectionnées au cas actuel. e.g., filtrage ou autres traitements des données, réglages des paramètres (ou coefficients). Les outils de Paramétrage de d'Adaptation des MÉthodes de Diagnostic aux Défauts (OPAMED2) réalisés sont expliqués dans le chapitre 4.
- Modélisation du robot et potentiellement des défauts selon les méthodes sélectionnées. En effet, les méthodes à base de modèle ont besoin du modèle cinématique et/ou dynamique du robot, et certaines méthodes à base de connaissances utilisent un modèle du défaut.

OMOPAMED2 permet la construction de la forme initiale de la MHD ( $MHD_0$ ). La distribution des méthodes de diagnostic intégrant la MHD risque de changer grâce à l'évaluation continue assurée par l'interaction avec l'opérateur via le Module d'Interaction Homme/Robot (MIHR).

# 3.2 Proposition d'une Méthode Hybride de Diagnostic : la MHD



## 3.2.1 Les étapes de la MHD

FIGURE 3.2 – Les éléments de la MHD dans son état n

Comme illustré à la figure 3.2, la MHD dans son état n (c.à.d. la forme de la MHD après la énième modification déclenchée par l'interaction avec l'humain) prend comme entrée les informations issues de la phase d'adaptation (défauts, méthodes adaptées, modèle du robot et modèles des

défauts) et les informations capteurs issues du robot. Les trois étapes principales constituant le module MHD sont :

- 1. Évaluation des méthodes de diagnostic : après l'identification des méthodes applicables pour chaque défaut, ces dernières subissent une évaluation à l'aide de l'indicateur de performance proposé dans 3.2.2.
- 2. Association des méthodes aux défauts : les meilleures méthodes pouvant isoler ou détecter chaque défaut sont identifiées. Un défaut peut être détecté au mieux à l'aide d'une méthode i et isolé au mieux à l'aide d'une méthode j. i.e. la méthode i est la plus précise en terme de détection, elle déclenche une alarme plus rapidement et elle a moins de fausses alarmes que la méthode j. Mais, la méthode j est plus précise en terme d'isolement. Une fois le défaut détecté, elle identifie sa source plus rapidement que la méthode i. Rien n'empêche que les méthodes i et j soient la même méthode. Grâce à cette évaluation, une première forme de la MHD est établie.
- 3. Mise en œuvre des méthodes sélectionnées : Les méthodes sélectionnées détectent et isolent les défauts en traitant les données et mesures robot ainsi qu'en se servant des modèles et des connaissances établies auparavant.

La suite de cette partie détaille la démarche proposée afin d'évaluer les méthodes de diagnostic utilisée dans la MHD.

# 3.2.2 Évaluation des méthodes de diagnostic

Afin d'évaluer objectivement les méthodes de FDI et les comparer entre elles d'une manière fiable et objective, un indicateur de performance générique est proposé dans cette section. Cet indicateur est composé de quatre sous-indicateurs reflétant respectivement la performance en détection, la performance en isolement, le coût d'application et la capacité de prévision . Le tableau 3.1 résume la structure de cet indicateur. Les éléments du tableau 3.1 sont classifiés

Sous-indicateur	Éléments
Détection	Taux de fausse alarme (FAR) [68]
	Taux de détection ratée (MDR) [68]
	Délai de Détection (DD)
	Stabilité de la Décision (SD)
	Taux de faux isolement (FIR)
Isolement	Délai d'Isolement (ID)
	Délai de rétablissement du mode normal(NMRD)
Coût	Temps de calcul par cycle (TPC)
	Volume de stockage requis (RDS)
	Nombre de défauts traités
Prévision	Diagnostic de défauts non prévus (NonPrev)
	Estimation d'Erreur (EE)
	Adaptabilité (Adap)

Tableau 3.1 – Sous indicateurs de l'indicateur de performance

en deux catégories : 'favorisant la détection rapide' soulignés dans ce tableau, et 'favorisant la robustesse contre les fausses alarmes' mis en gras.

Ces éléments sont choisis pour capter les caractéristiques les plus importantes. D'autres éléments peuvent être ajoutés, comme le temps moyen hors de contrôle, mais il s'agira d'informations redondantes.

La mise en application de cet indicateur demande de suivre ces étapes :

- 1. Définir l'importance de chaque sous élément en se reposant sur les préférences de l'utilisateur qui dépendent de la nature de l'application;
- 2. Calculer les éléments des sous-indicateurs;
- 3. Normaliser et calculer chaque sous-indicateur en effectuant une somme pondérée.

**Définition de l'importance des éléments** Selon l'application, un utilisateur peut s'intéresser à des éléments des sous indicateurs de détection et d'isolement plus que d'autres. Par exemple, le bon fonctionnement du GPS est très important dans une application de suivi par GPS. Dans ce cas, nous avons intérêt à isoler rapidement tous les défauts GPS qui peuvent survenir ; la présence de quelques fausses alarmes pourrait être tolérée contrairement aux autres types de défaut qui peuvent surgir sur des capteurs dont on a pas forcément besoin dans cette application.

Calcul des éléments Les équations (3.1) sont utilisées pour calculer FAR, MDR et FIR.

$$FAR = \frac{Faux \text{ positifs}}{Données \text{ sans défaut}}$$
$$MDR = \frac{Faux \text{ négatifs}}{Données \text{ avec défaut}}$$
$$FIR = \frac{Faux \text{ isolement}}{Donnés \text{ avec défaut}}$$
(3.1)

La capacité de la méthode à détecter que le mode normal est rétabli après apparition d'un défaut est une caractéristique importante. Ainsi, l'élément du sous-indicateur d'isolement "délai de rétablissement du mode normal" correspond au temps moyen requis par la méthode FDI pour reconnaître le mode normal après un défaut. Si n est le nombre de fois où le mode normal est rétabli, NMRD est calculé comme suit :

$$NMRD = \sum_{0}^{n} \frac{\text{Temps de retablissement}}{n}$$
(3.2)

L'élément "stabilité de la décision" (SD), un sous-indicateur de détection, mesure les oscillations faites par la sortie de la méthode. L'équation 3.3 décrit la méthode de calcul de cet indicateur.

$$SD = \frac{\sum_{0}^{n} \text{distance}(D_{t} - D_{t-1})}{n}$$
(3.3)

Avec,  $D_t$  et  $D_{t-1}$  respectivement la décision à l'instant t et t-1.

Les sous éléments DD et ID correspondent au temps moyen par itération mis par la méthode, respectivement pour détecter et isoler le défaut.

Les éléments des sous-indicateurs coût et prévision sont facilement déterminables : TPC est le temps moyen de calcul consommé par la méthode à chaque itération. RDS est le volume requis de données(en MO); le "nombre de défauts traités" est le nombre de défauts que la méthode peut isoler en même temps. EE et "Diagnostic de défauts non prévus" sont des valeurs booléennes représentant respectivement la capacité de la méthode à produire une estimation sur la fiabilité de sa décision et à détecter les défauts non prévus au départ. Finalement, l'élément "adaptabilité" est une estimation empirique de la facilité d'adaptation d'une méthode à d'autres robots ou d'autres défauts.

**Normalisation et calcul** La valeur d'un sous-indicateur correspond à la somme pondérée de ses éléments normalisés.

La normalisation des éléments calculant des durées (DD, ID et NMRD) se fait à l'aide de la fonction définie dans l'équation 3.4. Cette fonction garantie une valeur normalisée entre 0 et 1.

Valeur normalisée = 
$$e^{-0.5 \text{délai}}$$
 (3.4)

En ce qui concerne les éléments représentant des taux (FAR, FID et MDR), la valeur normalisée est simplement 1 - x. Ce qui donne une valeur élevée pour des taux faibles et inversement.

L'élément 'stabilité de la décision' ne rentre pas dans le calcul du sous-indicateur directement. Il sera utile uniquement pour la comparaison entre deux méthodes au cas où elles ont une évaluation égale.

Finalement, la valeur finale du sous-indicateur est obtenue par la somme pondérée des éléments comme il est montré dans l'équation 3.5.

Sous-indicateur = 
$$\frac{\sum (\text{Élément.Poids})}{\text{Nombre d'éléments}}$$
 (3.5)

Le poids correspondant à chaque élément est déterminé lors de la phase de définition de l'importance des éléments.

Au final, chaque sous-indicateurs aura une valeur située entre 0 et 1. Les équations suivantes résument la démarche aboutissant au calcul de l'indicateur :

$$\phi_{\text{Eval méthode}} = \frac{a * \phi_{\text{Détection}} + b * \phi_{\text{Isolement}} + c * \phi_{\text{Coût}} + d * \phi_{\text{prévision}}}{a + b + c + d}$$
(3.6)

avec  $\phi_x$  la valeur du sous-indicateur x; a, b, c et d sont des poids à définir selon la tâche. Par exemple, si dans une application la détection d'un défaut est cruciale, l'isolement est un peu moins important que la détection, le coût est d'importance moyenne et la prévision n'est pas importante, cela se traduit par (a, b, c, d) = (2, 1.5, 1, 0.5). Les  $\phi_x$  valeurs sont calculées de la manière suivante :

$$\phi_{\text{Détection}} = \frac{\alpha_0 * (1 - \text{FAR}) + \beta_0 * (1 - \text{MDR}) + \gamma_0 * e^{-0.5 * \text{DD}} + \sigma * (1 - \text{SD})}{4}$$

$$\phi_{\text{Isolement}} = \frac{\alpha_1 * (1 - \text{FIR}) + \beta_1 * e^{-0.5 * \text{ID}} + \gamma_1 * e^{-0.5 * \text{NMRD}}}{3}$$

$$\phi_{\text{Coût}} = \frac{\alpha_2 * e^{-40 * \text{TPC}} + \beta_2 * e^{-0.01 \text{RDS}} + \gamma_2 * \frac{N_{\text{défauts considérés}} - N_{\text{défauts traités}}}{N_{\text{défauts considérés}}}}{3}$$

$$\phi_{\text{Prévision}} = \frac{\alpha_3 * \text{NonPrev} + \beta_3 * EE + \gamma_3 * \frac{Adapt}{10}}{3}$$
(3.7)

Avec, les  $\alpha, \beta, \gamma$  et  $\sigma$  des poids choisis au moment de l'implémentation de l'indicateur selon l'importance des éléments.

Les éléments représentant un taux sont normalisés simplement par la fonction (1 - x). Par conséquent, si les taux sont nuls (FAR=0 par exemple) la valeur normalisée associée sera maximisée (1).

Les valeurs représentant des délais (e.g., DD) sont normalisées par la fonction  $e^{-0.5*x}$  dont la valeur 0.5 est choisi empiriquement pour que la valeur normalisée soit maximale (1) si le délai est nul et elle décroît raisonnablement avec l'augmentation du délai, e.g., pour un délai de 1s, la valeur associée est 0.6.

TPC (en secondes) et RDS (en MO) sont normalisés par des fonctions similaires mais avec un paramètre près de différence assurant une normalisation logique de chacun. La valeur normalisée de TPC vaut 0.45 pour un TPC de 0.02s, elle vaut 0.9 pour un TPC de 0.003. La valeur normalisée de RDS vaut 0.13 pour un RDS de 200MO, elle vaut 0.9 pour un RDS de 10MO.

# 3.3 Proposition d'un module d'interaction homme robot



FIGURE 3.3 – MIHR

Suite à la décision prise par la MHD, le Module d'Interaction Homme/Robot (MIHR) intervient pour assurer une interaction optimale avec l'humain ainsi qu'une amélioration continue de la MHD.

Le schéma du MIHR est présenté à la figure 3.3. Grâce à l'algorithme de raisonnement à partir de cas (CBR) intégrant une Base de Connaissance (BdC), MIHR propose des solutions pour les défauts isolés et aide l'opérateur à isoler en cas de défaut détecté et non isolé par la MHD. Une interface modulable s'occupe de l'interaction avec l'humain pour lui fournir les informations pertinentes et pour lui permettre de proposer des solutions et des isolements. La figure 3.4 montre le fonctionnement de la MIHR. Elle peut rencontrer plusieurs situations durant son fonctionnement. En ce qui concerne le diagnostic d'un défaut, nous distinguons trois cas : défaut isolé, défaut détecté non isolé et défaut signalé par l'opérateur non détecté par le système. Pour chacun de ces trois cas, une démarche différente est appliquée.



FIGURE 3.4 – Forme globale de la MIHR en application

## 3.3.1 Défaut isolé

Un défaut peut être isolé directement par la MHD ou bien à l'aide de l'expertise de l'opérateur. Dans les deux cas, suite à son isolement, une étude de criticité est faite. La criticité d'un défaut dépend de quatre éléments [74] :

- 1. Importance du composant : Selon la mission du robot, il peut s'appuyer sur des composants (capteurs) plus que sur d'autres. Plus le robot compte sur un composant pour réussir sa tâche, plus ce composant est important. Par exemple, dans une application de suivi de trajectoire par GPS, le GPS est plus important que les encodeurs des roues. En conséquence, un défaut sur le GPS sera considéré plus critique qu'un défaut sur les encodeurs.
- 2. Amplitude du défaut : L'influence du défaut sur le comportement du robot est considérée lors du calcul de criticité. Par exemple, une erreur d'écart de 20 cm sur la mesure de position est plus grave qu'une erreur de 5cm.
- 3. Fréquence de l'alarme : Lorsque le défaut se répète, il sera considéré comme plus grave qu'un défaut ponctuel.
- 4. Niveau de confiance de la décision : Certains méthodes de diagnostic nous permettent d'avoir le niveau de confiance de sa décision. Si c'est le cas, cet élément est pris en compte

lors de l'étude de criticité. Un niveau de confiance plus bas implique un niveau de criticité plus bas en cas de défaut.

**Étude de criticité** Pour conclure sur l'importance de l'information de diagnostic directement proportionnel à la criticité du défaut, la théorie de Dempster-Shafer (DS) [74] est appliquée. Cette théorie permet de fusionner des informations pour produire une conclusion sur la situation. La formule de base est définie comme dans l'équation (3.8) connue en tant que règle de combinaison de Dempster qui est une généralisation du théorème de Bayes. Cette règle ignore les conflits grâce à un facteur de normalisation et met en valeur l'accord entre des sources multiples.

$$m(A) = \frac{1}{1-k} \sum_{B_i \wedge C_i \wedge D_i \dots = A} m_1(B_i) . m_2(C_j) . m_3(D_l) \dots$$

$$k = \sum_{B \wedge C \wedge D \dots = \emptyset} m_1(B_i) . m_2(C_j) . m_3(D_l) \dots$$
(3.8)

Dans cette équation, m(A) est la masse d'un élément A de l'ensemble des parties (ex : A peut faire référence à l'état "Très grave" ou "négligeable"). Elle exprime la proportion de toutes les preuves affirmant que A est l'état actuel. Par définition, la masse de l'ensemble vide vaut zéro  $m(\emptyset) = 0$ , et la somme des masses de tous les sous-ensembles vaut  $1 \sum_{\Omega} m(A) = 1$ . c.à.d, la probabilité que l'état actuel ne soit aucun des états possibles est nulle, et la probabilité que l'état actuel soit l'un des cas possibles est 1. k est la mesure du niveau de conflit entre les masses. Le facteur de normalisation 1 - k permet d'ignorer les conflits et d'attribuer toute masse impliquée dans un conflit à 0. En conséquence, cela donne des résultats contre-intuitifs face à des conflits significatifs, ce qui n'est pas notre cas.

Nous définissons quatre niveaux de criticité :

- 1. Faible : Le défaut est négligeable, il n'influence pas la mission du robot.
- 2. Moyen : Le défaut peut perturber le robot sans forcément faire échouer la mission.
- 3. Haut : Le défaut peut faire échouer la mission du robot.
- 4. Fatal : Le défaut est dangereux, il peut endommager le matériel ou rendre le robot incontrôlable.

À chaque type de défaut, nous choisissons les valeurs des masses correspondant à chaque niveau de criticité comme dans le tableau 3.2.

Type du défaut	Masses associé aux états de criticité		
	(Fatal Haut Moyen Faible)		
Mécanique bloquant	$\begin{pmatrix} 0.8 & 0.1 & 0.05 & 0.05 \end{pmatrix}$		
Capteur principal	$\begin{pmatrix} 0.8 & 0.1 & 0.05 & 0.05 \end{pmatrix}$		
Capteur important	(0.1  0.8  0.07  0.03)		
Capteur secondaire	(0.03  0.07  0.8  0.1)		
Capteur non utilisé	(0.01  0.01  0.18  0.8)		
Inconnu	(0.25  0.25  0.25  0.25)		

Tableau 3.2 – Exemple de masses associées aux états de criticité selon le type de défaut

Un défaut mécanique bloquant endommage le matériel. Pour cela, la masse de l'état de criticité fatal est élevée.

Le capteur principal est, s'il existe, le capteur sur lequel s'appuie le robot pour réussir sa tâche. Dans le cas du suivi de trajectoire par GPS par exemple, le GPS est le capteur principal. Une perte de ce dernier engendrera l'échec de la mission.

Un capteur important est un capteur utilisé pour avoir des informations importantes, mais il n'est pas primordial comme le capteur principal. Prenons aussi le cas du suivi de trajectoire par GPS, l'IMU est utile pour connaître l'orientation du robot afin de contrôler le cap du robot. Toutefois, l'orientation est estimable par d'autres moyens. Si cette estimation n'est pas prévue, la mission du robot risque d'échouer au cas de défaut sur l'IMU. Pour cela, la masse de l'état de criticité haut est importante.

Un capteur secondaire possède moins d'effet sur la tâche qu'un capteur important. Dans le même exemple de suivi, un laser peut être utile pour éviter les obstacles. Mais, au cas où le robot se trouve momentanément dans un endroit sans obstacles, ce capteur n'a pas trop d'importance. À ce moment, le laser est un capteur secondaire. La masse de l'état de criticité moyen est importante.

Certains capteurs peuvent être ignorés lors d'une application spécifique. Un GPS par exemple n'est pas utilisé pendant une mission de suivi de structure. Dans ce cas, un défaut sur ce capteur n'affecte en rien la réussite de la mission.

Dans le cas d'identification d'un nouveau défaut non prévu, les masses de tous les états de criticité sont égales.

Quant au niveau de confiance, les masses correspondant à l'état de criticité sont tirées des valeurs de l'indicateur de performance associé (section 3.2.2), exploitées par de la logique floue. Plus la méthode qui a prédit la présence de défaut est fiable, plus cette information est importante. Par exemple, si une méthode ayant un taux de fausses alarmes de 0.9 (score normalisé proche du zéro), sa décision est beaucoup moins importante qu'une méthode possédant un taux de fausses alarmes de 0.05 (score normalisé proche de 1).

La figure 3.5 montre la variation des fonctions d'appartenance aux états de criticité selon la valeur de l'indicateur correspondant. Lorsque la valeur de l'indicateur de performance augmente de 0.8 à 1, la masse correspondant à l'état fatal s'accroît de 0 à 1 et celle correspondant à l'état haut diminue de 1 à 0. Les fonctions mathématiques associées sont énoncées dans les équations :

$$m_{\text{fatal}} = \begin{cases} 5x - 4 & \text{si } x > 0.8 \\ 0 & \text{sinon.} \end{cases}$$

$$m_{\text{haut}} = \begin{cases} -5x + 5 & \text{si } x \in [0.8, 1] \\ 5x - 5 & \text{si } x \in [0.6, 0.8] \\ 0 & \text{sinon.} \end{cases}$$

$$m_{\text{moyen}} = \begin{cases} -5x + 4 & \text{si } x \in [0.6, 8] \\ 5x - 2 & \text{si } x \in [0.4, 0.6] \\ 0 & \text{sinon.} \end{cases}$$

$$m_{\text{faible}} = \begin{cases} -5x + 3 & \text{si } x \in [0.4, 6] \\ 1 & \text{si } x < 0.4 \\ 0 & \text{sinon.} \end{cases}$$
(3.9)

x représente dans ces équations la valeur de l'indicateur associé,  $x \in [0, 1]$ .



FIGURE 3.5 – Les fonctions d'appartenance des états de criticité

**Proposition d'une solution** Suite à l'isolement d'un défaut et à l'étude de criticité, une solution est proposée. Selon le niveau de criticité, cette solution pourra être appliquée directement ou bien après la validation de l'opérateur qui aura le droit d'accepter ou de refuser la solution proposée par l'algorithme. Dans tous les cas, l'opérateur évalue la pertinence de la solution appliquée. Si la solution n'est pas adéquate, il peut en proposer une autre ou bien déclarer un défaut détecté non isolé. La base de connaissances est alors mise à jour conformément à cette évaluation pour en profiter dans les prochaines décisions.

L'algorithme de raisonnement à partir de cas (en anglais Case-Based Reasoning (CBR))est efficace pour ce genre de problèmes [17]. Cet algorithme copie le comportement humain qui consiste à faire naturellement appel à l'expérience pour résoudre les problèmes en se souvenant des situations semblables rencontrées précédemment et en les comparant à la situation actuelle dans le but de construire une nouvelle solution qui, après sa validation, s'ajoutera à l'expérience. Dans le cas d'un CBR dont le but est de trouver une solution à un cas (ou situation), le cycle de travail de cet algorithme est composé de quatre étapes principales :

- 1. La recherche des cas similaires : Pour cela, il faut des règles pour définir la similarité entre les cas, i.e., une méthode pour déterminer la métrique entre les cas;
- 2. Réutilisation : Après avoir identifié le cas le plus similaire, utiliser ce cas comme solution du cas actuel. Une phase d'adaptation peut être faite aussi avant l'utilisation du cas;
- 3. Révision : Vérifier si la solution candidate correspond bien au cas actuel. Si la solution est validée, on procède à son application, sinon soit un autre cas est choisi soit l'humain propose sa propre solution;

4. Mise à jour de la base de données : Appelée aussi l'étape d'apprentissage, cette étape consiste à ajouter le retour d'expérience à la base de données. Le cas (les données), la solution proposée et l'évaluation obtenue dans l'étape de révision sont ajoutés à la base de données pour l'enrichir.

Ces systèmes utilisent de nombreux types de connaissances. L'auteur de [66] identifie quatre conteneurs (types) de connaissances : le vocabulaire, les mesures de similarité, les connaissances d'adaptation et les cas eux-mêmes, tels que :

- Le vocabulaire : Il comprend les variables nécessaires pour décrire les cas. Ces variables doivent être spécifiées de manière à satisfaire à la fois deux conditions : 1- être utiles pour récupérer d'autres cas, qui contiennent des solutions utiles à des problèmes similaires, et 2- être suffisamment discriminantes pour empêcher la récupération de cas trop différents, ce qui pourrait conduire à de fausses solutions et dégrader la performance. Pour bien choisir ces variables, une bonne connaissance sur le système est primordiale.
- La mesure de similarité : Elle définit la méthode par laquelle nous comparons la similarité de deux cas. Il existe différentes solutions à un problème. Toutefois, il faut choisir une méthode adéquate selon le nombre de variables constituant un cas et les types de données enregistrées.
- L'adaptation : Elle comprend les connaissances nécessaires à la mise à jour continue de la CBR. C'est la réponse à la question : comment profitons-nous du retour d'expérience de l'opérateur ? Cette étape peut être confiée à l'opérateur qui décidera de l'influence du cas actuel sur les futures décisions.
- Les cas : Ils représentent la connaissance que le système acquiert pendant l'utilisation ainsi que les connaissances de départ enregistrées dans la base de données.

Un premier CBR est proposé pour la proposition d'une solution au cas d'un défaut isolé. Les éléments de ce CBR sont les suivants :

- Vocabulaire : Un cas est composé de la décision finale de la MHD ainsi que de plusieurs informations utiles servant à bien représenter et discriminer les cas :
  - Les valeurs des indicateurs de performance de toutes les méthodes contribuant à cette décision, c.à.d., les sous-méthodes de diagnostic pouvant diagnostiquer le statut actuel;
  - Les résidus des sous-méthodes pouvant diagnostiquer l'état actuel, c.à.d., les résidus qui captent la présence du défaut diagnostiqué;
  - Les valeurs des sous-indicateurs de performance correspondant à ces sous-méthodes;
  - Les informations complémentaires sur le défaut : sa localisation (mécanique, capteur ou extérieur), sa nature (perte, étape, perturbation ou dérive) et sa sévérité;
  - La solution proposée est sauvegardée aussi avec son évaluation qui peut être +1 si cette solution a été jugée adéquate ou -1 sinon;
- Mesure de similarité : Pour retirer les cas les plus similaires, la mesure de similarité doit être à la fois rapide et précise. Vu la dimension de données et leurs différents types, il vaut mieux éviter les méthodes de classification classique de type K plus proches voisins. La méthode appliquée consiste à associer à chaque solution présente dans la base de données, un score en se basant : 1- sur la ressemblance entre les cas (dans la base de données) contenant cette solution et le cas actuel et 2- sur l'évaluation de ces cas. En

effet, le score final pour une solution s est présenté dans l'équation :

$$\operatorname{score}(s) = \frac{\sum_{\text{Base de données}} \operatorname{Sim}(X, Y_i) * e_i}{N_s}$$
(3.10)

Avec  $e_i$  l'évaluation du cas  $Y_i$  présent dans la base de données et  $N_s$  le nombre des données  $(Y_i)$  dans la base ayant contribué au calcul du score (s); Sim(X, Y) est la mesure de similarité entre les deux cas X (actuels) et  $Y_i$  (présent dans la base de données), sa formule est :

$$\operatorname{Sim}(X,Y) = \frac{\sum_{i} w_{i} \operatorname{sim}(x_{i}, y_{i})}{\sum_{i} w_{i}}$$
(3.11)

Avec  $w_i$  le poids d'un élément du vecteur de données. Pour le calculer, nous appliquons la formule  $w_i = 1 - \frac{rang_i - 1}{max(Rang)}$  avec  $rang_i$  le rang attribué à un élément *i* après le rangement de tous les éléments par ordre décroissant. Le tableau 3.3 montre les rangs et le poids

Élément		poids
Localisation du défaut	1	1
Nature du défaut	1	1
Criticité du défaut	1	1
Décision de la MHD	1	1
Décision des sous-méthodes	2	2/3
Taux de détection des sous-méthodes impliquées	2	2/3
Taux d'isolement des sous-méthodes impliquées	3	1/3

Tableau 3.3 – Rangs et poids des informations dans le CBR de prédiction de solutions

associés à chaque élément.

Quant à la mesure locale de similarité sim(x, y), celle-ci est adaptée au type des éléments comparés :

- Les valeurs de l'indicateur de performance et la mesure de criticité (Dempster) sont des valeurs numériques. Leur similarité est mesurée par la distance euclidienne entre les deux valeurs comparées.
- La distance entre deux décisions prises par la MHD ou deux décisions prises par une sous-méthode vaut 0 si les deux décisions sont identiques, 0.5 si les deux décisions appartiennent au même groupe de statuts (e.g. chaque méthode détecte le défaut mais au mauvais endroit), 1 sinon.
- Pour les données alphabétiques, la mesure de la distance est binaire (0 si deux données identiques, 1 sinon) sauf pour la similarité entre la nature des défauts qui est tirée du tableau 3.4. Dans ce tableau, la distance entre deux défauts de natures très différentes vaut 1. Elle est nulle si les deux natures comparées sont identiques. La dérive et la perturbation sont plus semblables entre eux que le biais et la dérive, pour cela, la distance attribuée entre les deux premiers est 0.3, pour les deux derniers elle vaut 0.5.

Ainsi, pour chaque solution enregistrée dans le CBR, cet algorithme associera un score. Plus le cas actuel ressemble à un cas évalué positivement dans la base de données, plus la solution adoptée dans ce cas sera favorisée. Et inversement, plus le cas actuel ressemble à un cas évalué négativement dans la base de données, plus la solution adoptée dans ce cas sera défavorisée. La solution possédant le score le plus élevé sera proposée.

Nature	Dérive	Step	Perte	Perturbations
Dérive	0	0.5	1	0.3
Step	0.5	0	1	0.3
Perte	1	1	0	1
Perturbations	0.3	0.3	1	0

Tableau 3.4 – Distance entre les natures des défauts adoptée pour le calcul de similarité

- L'adaptation : Cette étape consiste dans notre cas à sauvegarder dans la base de données associée, le cas actuel ainsi que la décision prise par l'opérateur concernant la solution proposée, i.e., admission ou rejet de la solution proposée. Si l'opérateur refuse la solution proposée, le cas sera sauvegardée avec une évaluation de -1. L'opérateur pourra alors proposer une solution qui, si elle est validée, sera enregistrée dans la base de données dans un nouveau cas évalué positivement. Nous décrivons, plus tard dans ce chapitre, comment l'opérateur peut proposer une solution.
- Les cas : La base de données contient au départ les règles de bases issues de l'expertise, avec des évaluations positives généralement.

Ainsi, ce CBR aide l'opérateur à proposer des solutions d'une manière intelligente ressemblant au raisonnement humain. De plus, il apprend et s'améliore continuellement tout en interagissant avec l'opérateur considéré comme expert. Finalement, la solution proposée sera appliquée.

## 3.3.2 Défaut détecté non isolé

Dans le cas où un défaut est détecté mais n'est pas isolé, le mode d'aide à la décision est activé. Les informations pertinentes sont extraites de la MHD. Elles sont envoyées à l'opérateur pour l'aider à comprendre la situation et à isoler le défaut. Un autre CBR est construit pour l'aide à l'isolement. Son algorithme est identique à celui du CBR d'aide à la proposition de solutions décrit dans le paragraphe précédent, seul le vocabulaire diffère. Le but de cet algorithme étant d'aider l'opérateur à isoler le défaut et non plus de proposer une solution, le calcul de similarité prend en compte l'évaluation de l'isolement proposé au lieu de celle de la solution. Le vocabulaire de ce CBR est composé de la décision de la MHD, de la décision des différentes méthodes de diagnostic et des valeurs de l'indicateur de performance de toutes les sous-méthodes.

La définition des cas de départ est plus compliquée dans ce CBR. Il est envisageable de les définir grâce à une pré-simulation dans des conditions optimales. La base de données s'enrichira grâce à l'interaction avec l'humain. Cette étape vise essentiellement à proposer un isolement dans le cas où les sous méthodes de la MHD n'arrivent pas à trouver la source du défaut ou bien lorsque les taux de confiance de celles-ci ne sont pas acceptables.

Si l'isolement du défaut est réussi, la démarche décrit dans 3.3.1 est entamée pour proposer une solution. Sinon, l'opérateur décide s'il faut arrêter le robot ou pas. Une intervention d'un expert sera nécessaire. Toutefois, les informations stockées de la MHD et des capteurs sont consultables.

#### 3.3.3 Ajustement des sous-méthodes

Grâce à l'évaluation faite par l'opérateur, les valeurs des indicateurs de performance associés sont ajustées. En effet, l'historique d'évaluation est sauvegardé sur une fenêtre glissante de taille n(dans notre cas n = 100). Une fois tous les n itérations, les valeurs de l'indicateur de perfor-

mance des méthodes sont ajustées en se basant sur cet historique.

Considérons une méthode m traitant un défaut d. Soit  $G = (g_1, g_2, ..., g_n)$  l'historique des bonnes décisions tel que  $g_k = 1$  si, à l'itération  $t_{\text{actuel}} - k$ , la décision prise a été évaluée comme bonne décision (défaut d bien isolé),  $g_k = 0$  sinon; de la même manière,  $Fa = (fa_1, fa_2, ..., fa_n)$ l'historique des fausses alarmes,  $Fi = (fi_1, fi_2, ..., fi_n)$  l'historique des faux isolements et  $Md = (md_1, md_2, ..., md_n)$  l'historique des détections ratées. L'effet de chaque phénomène (bonne décision, fausse détection, faux isolement et détection ratée) est calculé comme dans l'équation :

$$e(G) = +\sum_{i} e^{-i\lambda} g_{i}$$

$$e(Fa) = -\sum_{i} e^{-i\lambda} fa_{i}$$

$$e(Fi) = -\sum_{i} e^{-i\lambda} g_{i}$$

$$e(Md) = -\sum_{i} e^{-i\lambda} md_{i}$$
(3.12)

Avec  $\lambda$  le facteur d'oubli. Il sert à atténuer l'effet des anciens points.

Ensuite, la valeur de l'indicateur  $\tau$  de la méthode m associée au défaut d est ajustée de cette manière :

$$\tau_k = \tau_{k-1} + \frac{e(G)}{2} + e(fa) + \frac{e(fi)}{2} + e(md)$$
(3.13)

Si la performance d'une méthode est devenue faible, un ajustement automatique est envisageable. Par exemple, si une méthode se basant sur le seuillage (comme le KF) souffre d'un taux élevé de fausse alarme, une augmentation du seuil pourrait les réduire.

#### 3.3.4 Défaut signalé par l'opérateur

Au cas où l'opérateur constate un défaut et que la MHD n'a rien constaté, la même démarche que celle décrite pour un défaut détecté non isolé (dans 3.3.2) est suivie, i.e., le mode d'aide à la décision est activé. D'abord, le CBR prédit un défaut qui sera validé ou refusé par l'opérateur. Ensuite, le mode d'aide à la proposition de solutions est activé en se basant sur la décision prise dans l'étape d'isolement. Finalement, les retours d'expérience sont enregistrés dans les bases de données.

**Nature du défaut** L'étude pour déterminer la nature du défaut n'est pas encore faite. Nous l'avons considérée car, en réalité, la solution attribuée à un dysfonctionnement varie selon la nature du défaut. Par exemple, un défaut de biais statique sur la mesure de l'IMU peut être induit par une fausse calibration tandis qu'un défaut de perturbation sur ce signal peut être induit par une mauvaise fixation sur le robot.

#### 3.3.5 L'interaction avec l'humain

Modes et transitions de point de vue de l'opérateur Comme décrit dans la section 3.3, la méthode proposée interagit avec l'opérateur humain en temps réel. L'humain possède trois missions essentielles :

1. Supervision : superviser le robot, et ceci dans tous les cas;

- 2. Contribution à l'isolement du défaut : isoler le défaut en cas de défaut détecté non isolé et évaluer l'isolement en cas de défaut isolé;
- 3. Contribuer à la proposition d'une solution : évaluer la solution proposée par la MHD et proposer une solution dans le cas échéant.

La figure 3.6 montre les transitions entre les différents modes du point de vue de l'opérateur ainsi que les transitions possibles.



FIGURE 3.6 – Les modes et les transitions du point de vue opérateur

Initialement le mode surveillance est actif. Dans ce mode, l'opérateur peut suivre les informations de base, i.e., la position du robot, sa vitesse et la commande. Toutefois, l'opérateur a le droit de demander plus d'informations qui sont classées en trois catégories : Les résidus captant les défauts, la mesure des capteurs et la valeur des indicateurs de performances.

À tout moment, l'opérateur peut signaler un défaut. Dans ce cas, le mode d'aide à l'isolement est activé. Il isole le défaut précis ou prédit seulement le type et la nature du défaut. Ensuite, une solution est proposée par la MHD automatiquement et le mode d'aide à la proposition de solution est activé. Selon le niveau de criticité du défaut, l'application de la solution exige la validation de l'opérateur.

**Principes pour construire une interface graphique de communication Homme-machine** L'interface graphique de communication Homme-machine que nous proposons obéit à certains critères inspirés de la norme [3] et des propositions formulées dans [81]. Généralement, la conception passe par trois phases principales :

1. Analyse : cette phase consiste à identifier le cahier des charges. Notamment, l'objectif de l'interface, les besoins de l'opérateur, les informations disponibles et les informations à échanger. Idéalement, l'analyse est achevée en se basant sur la recherche bibliographique, sur des entretiens avec les opérateurs et par observations.

- 2. Conception : cette phase consiste à concevoir un premier prototype répondant au cahier des charges. Le but en est toujours de faciliter la communication et d'améliorer la compréhension.
- 3. Implémentation, test et validation : L'interface conçue est implémentée et testée par un groupe d'opérateurs. Le but est d'avoir un retour d'expérience pertinent afin de corriger les erreurs éventuelles dans la conception.

Les principes de bases que nous avons considérés pour la conception visent essentiellement à assurer une communication efficace tout en présentant les informations nécessaires d'une bonne manière et en réduisant le temps de réaction.

- 1. Anticipation des besoins de l'opérateur : les fonctions les plus demandées par l'opérateur sont mises en avant. Ceci est possible grâce à la phase d'analyse qui permet d'identifier les attentes de l'opérateur.
- 2. Conformité à l'environnement physique : l'interface reflète l'environnement physique surveillé. Par exemple, si l'opérateur s'intéresse aux déplacements d'un robot dans une carte connue, l'interface doit bien montrer la position du robot par rapport à cette carte.
- 3. Organisation : les informations sont groupées et séparées d'une façon logique et compréhensible selon leurs finalités. Par exemple, les informations visant à localiser le robot sont séparées de celles dont le but est de connaître son état.
- 4. Minimisation de la charge cognitive : pour réduire la charge cognitive, seul les informations utiles et non redondantes sont affichées. Les informations les plus importantes sont les plus visibles. Un langage simple et compréhensible par l'opérateur est adoptée.
- 5. Statut et feedback : l'opérateur reste au courant des actions menées, des changements d'états et des erreurs qui peuvent l'intéresser. Pour que l'opérateur se repère rapidement dans l'interface, chaque fenêtre (ou sous-interface) possède une icône unique.
- 6. Respect des conventions : le même type d'interaction est adopté partout (e.g., liste défilante ou tableau). Les conventions des couleurs aussi sont respectées, e.g., les messages d'erreur s'affichent en rouge.
- 7. Tolérance aux erreurs : si l'opérateur commet accidentellement une erreur, il pourra la corriger facilement.

# 3.4 Conclusion

MHD et MIHR constituent ensemble le S2D2R qui assure le diagnostic en ligne du robot pendant qu'il effectue sa mission. Les sous-méthodes intégrées dans la MHD sont présélectionnées puis paramétrées hors ligne avant d'être évaluées, à l'aide de l'indicateur de performance développé dans 3.2.2. Cette évaluation mène une association optimale entre les méthodes de diagnostic et les défauts, ce qui permet le diagnostic des défauts en temps réel. La sortie de la MHD qui est le statut du robot (normal, défaut détecté ou défaut isolé) fait l'objet d'une entrée pour le Module d'Interaction Homme/Robot (MIHR) qui est expliqué dans 3.3. Il est composé de deux éléments principaux, à savoir, l'interface graphique et l'algorithme de raisonnement à partir de cas. Dans le chapitre suivant, le bloc OPAMED2 (figure 3.1) est détaillé, i.e., le chapitre suivant développe les contributions faites au niveau des méthodes de diagnostic et des données d'entrée pour les adapter pour le traitement des défauts en robotique mobile.

# Chapitre 4

# Ajustement et optimisation des méthodes de diagnostic de défauts en robotique mobile

Les méthodes de diagnostic doivent être adaptées aux défauts de la robotique mobile avant de les considérer dans la MHD. L'adaptation consiste à :

- Définir les modèles pour les méthodes à base de modèle, les règles ou les informations connues pour les méthodes à base de connaissances et l'architecture et les données d'entraînement pour les méthodes à base de données;
- Définir le prétraitement nécessaire des données dans certains cas;
- Régler les différents paramètres (gains, coefficients) présents dans les méthodes.

L'ensemble des contributions faites dans cette optique est détaillé dans ce chapitre.

## 4.1 Les filtres de Kalman

### 4.1.1 Définitions

Introduit en 1960 [44], le Filtre de Kalman (KF) est une technique d'estimation optimale. Il est largement utilisé dans les applications robotiques mobiles (par exemple pour la localisation [27], la fusion de données capteurs [6], ...). Son mode de fonctionnement est le suivant :

0. Initier la moyenne et la matrice de covariance des états.

Puis, pour chaque étape pas du temps :

- 1. Prédire l'état et sa matrice de covariance d'erreur.
- 2. Corriger les valeurs prédites par les mesures des capteurs.

Soit un système linéaire qui a la représentation d'état suivante :

$$\begin{cases} \dot{X}(t) = FX + Bu + v \\ Y(t) = HX + w \end{cases}$$
(4.1)

Où v et w sont des bruits gaussiens centrés et indépendants qui représentent respectivement les bruits sur le processus et celui sur la mesure. Ces derniers ont comme matrices de covariance respectives Q et R. F, H et B sont respectivement la matrice de transition d'état, la matrice d'observation, et la matrice optionnelle de contrôle. Le pseudo algorithme du KF est le suivant :

0. Définir les valeurs initiales des variables d'états et de la matrice de covariance de l'erreur :  $\hat{X}_0, P_0$ 

Puis, pour chaque pas de temps :

1. Prédiction de l'état et de la covariance de l'erreur :

$$\hat{X}_k^- = AX_{k-1}$$
$$P_k^- = AP_{k-1}A^T + Q$$

- 2. Calcul du gain de Kalman :  $\hat{K}_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$
- 3. Correction de la prédiction avec la mesure  $Z: \hat{X}_k = \hat{X}_k^- + K_k(Z_k h\hat{X}_k^-)$
- 4. Ajustement de la covariance de l'erreur :  $P_k = P_k^-(1 K_k H)$

Le paramétrage du KF consiste à définir  $X_0$  et  $P_0$ , ainsi que Q et R. Nous pouvons donner des valeurs exactes de  $X_0$  lors de l'initialisation, car elles sont généralement connues. En fait, plus les valeurs de  $P_0$  sont grandes, plus la variation de X sera importante dans l'étape suivante. R représente l'erreur de mesure, et Q représente celle du modèle. Plus les valeurs de R sont petites par rapport à celles de Q, plus le filtre se fiera aux mesures.

**EKF** Le KF standard est limité aux systèmes linéaires. L'EKF est une extension du KF pour outrepasser cette limitation. Il possède les mêmes étapes de prédiction/correction que le KF standard. En complément, une étape de linéarisation y est introduite. Les matrices d'observation et de transition sont remplacées dans l'algorithme par les Jacobiens :

$$A_{k} = \frac{\partial f}{\partial x} \Big|_{\hat{x}_{k|k}, u_{k}}$$

$$H_{k} = \frac{\partial h}{\partial x} \Big|_{\hat{x}_{k|k}}$$
(4.2)

Sachant que le modèle du système est :

$$x_{k} = f(x_{k-1}, y_{k-1}) + w_{k-1}$$
  

$$z_{k} = h(x_{k}) + v_{k}$$
(4.3)

L'EKF peut traiter de manière efficace des modèles non linéaires de premier ordre tout en ayant des niveaux de performance et d'efficacité proche de ceux d'un KF standard. Cependant, pour des modèles non linéaires d'ordre supérieur, la performance se dégrade (faible précision) et le temps de calcul sera plus élevé en raison du calcul lourd des matrices jacobiennes.

**UKF** Pour maintenir la performance et l'efficacité pour des systèmes non linéaires d'ordre supérieur à 1, l'UKF a été développé. Pour traiter des ordres supérieurs de non-linéarité (jusqu'à l'ordre 3) sans passer par des matrices jacobiennes, une transformation Unscented transform (UT) est faite. Elle consiste à calculer et transformer les points sigma (un ensemble de points déterministes choisis pour représenter les statistiques de la variable aléatoire) pour estimer la moyenne et la covariance du vecteur original. L'algorithme simplifié est le suivant [52] :

0. Initier la moyenne et la matrice de covariance :  $\hat{X}_0$  et  $P_0$ .

Puis, pour chaque pas de temps :

- 1. Calculer les points sigma et les poids (UT)
- 2. Prédire la moyenne et la matrice de covariance  $\hat{X}_k^-$  et  $P_k^-$  en utilisant les valeurs prédites
- 3. Corriger avec les mesures

**Calcul des résidus** La distance de Mahalanobis est une mesure de distance basée sur la corrélation entre deux variables pouvant identifier et analyser deux modèles [20]. La différence par rapport à la distance euclidienne repose sur le fait que la distance de Mahalanobis prend en compte la variance et la corrélation de la série de données. C'est une manière utile pour déterminer la similarité entre une série de données connues et une série de données inconnues ou bien pour déterminer la dissimilarité entre deux vecteurs aléatoires.

En pratique, la distance de Mahalanobis entre deux vecteurs aléatoires de données :

 $\vec{a} = (a_1, a_2, a_3, ..., a_n)^T$  et  $\vec{y} = (b_1, b_2, b_3, ..., b_n)^T$  possédant une matrice de covariance  $\Sigma$  est de la forme :

$$d(\vec{a}, \vec{b}) = \sqrt{(\vec{a} - \vec{b})\Sigma^{-1}(\vec{a} - \vec{b})}$$
(4.4)

Ainsi, cette distance peut être utilisée pour calculer un résidu à partir de n'importe quelle version du filtre de Kalman. Ceci est rendu possible par le calcul de similarité entre les données en entrée et les données prédites par le filtre.

Si S est la matrice de covariance de sortie du filtre de Kalman et y est le résidu simple  $y = (\hat{x} - x)$ avec x le vecteur d'état du modèle adopté et  $\hat{x}$  le vecteur d'état prédit, alors la distance de Mahalanobis entre le vecteur d'entrée et le vecteur des valeur prédites est de la forme :

$$r = yS^{-1}y^T \tag{4.5}$$

Plus la différence entre la mesure et la prédiction est importante, plus ce résidu sera grand. La détection des défauts se fera par seuillage de ce résidu.



FIGURE 4.1 – Algorithme génétique pour le paramétrage du filtre de Kalman

#### 4.1.2 Auto-paramétrage des filtres de Kalman

Comme indiqué précédemment, pour paramétrer un KF, il faut "régler" les matrices R et Q garantissant le bon fonctionnement du filtre. La matrice R peut être construite d'après les fiches techniques des capteurs. Elle représente l'incertitude des mesures. Si les valeurs d'états sont indépendantes, Q et R sont des matrices diagonales. Chaque élément de diag(R) est égal à la variance du bruit de la mesure correspondante. La matrice Q représente l'incertitude sur le

modèle. Chaque hypothèse de simplification prise lors de la conception du modèle affecte cette matrice. La méthode classique pour trouver cette matrice est celle de l'essai-erreur; ce qui ne garantit pas le bon choix. Pour fiabiliser et automatiser la construction de cette matrice, nous proposons une méthode basée sur un algorithme génétique.

Les algorithmes génétiques [34] sont des algorithmes évolutionnistes. Ils sont utilisés pour trouver une solution approchée à un problème de maximisation, de minimisation ou plus généralement d'optimisation. Pour trouver les paramètres qui optimisent une équation f, l'algorithme part d'une population initiale aléatoire dont chaque individu constitue une solution probable. Chaque individu est formé de plusieurs chromosomes dont chaque chromosome est un paramètre  $(x_0, x_1, ...)$ . Comme il est illustré à la figure 4.1, cet algorithme applique la démarche suivante :

- Sélection : Suite à une évaluation individuelle (insertion dans f), un poids (fitness) est associé à chaque individu. Une sélection aléatoire se fait ensuite afin de sélectionner un nombre prédéfini d'individus. Plusieurs techniques peuvent être utilisées pour faire cette sélection. Le point commun entre elles est que les individus qui ont le plus grand poids auront la plus grande chance d'être sélectionnés.
- 2. Croisement : Les individus choisis sont croisés deux à deux. Chaque croisement donne lieu à un nouvel individu dont les chromosomes sont la combinaison de ceux de ses parents. La probabilité d'apparition d'un chromosome de l'un ou l'autre des deux parents dans ceux de l'enfant dépend du paramétrage initial et de la technique de combinaison.
- 3. Mutation : Suivant un taux de mutation prédéfini généralement très faible (< 0.1), la valeur de chaque chromosome peut subir un changement. Cette étape sert à éviter une convergence prématurée.
- 4. Reproduction : cette démarche est reproduite sur les nouveaux individus jusqu'à la validité d'un des critères d'arrêt (nombre d'itérations, variation du poids maximum d'une étape à une autre, ...).
- 5. L'individu qui possède le poids le plus important est alors retenu comme solution du problème d'optimisation.

Dans notre cas d'étude, nous utilisons cet algorithme pour trouver la matrice Q minimisant une fonction de coût :

$$C = \begin{cases} |r - 0| \text{ dans le cas normal} \\ |r - h| \text{ dans le cas d'un défaut} \end{cases}$$
(4.6)

Où, r est le résidu généré en calculant la distance de Mahalanobis et h est une grande valeur à choisir empiriquement. L'algorithme d'auto-paramétrage se fait hors ligne sur une base de données. Dans notre cas, cette base de données contient les mesures (x, y) et la commande (v, w)ainsi qu'une indication sur la présence d'un défaut afin de pouvoir choisir la valeur de h. La population est un ensemble de valeurs possibles de diag(Q). Chaque individu est formé de 3 chromosomes  $(Q_{xx}, Q_{yy}, Q_{zz})$ . L'algorithme devient donc :

- 1. Initialiser les différents paramètres de l'algorithme.
- 2. Pour chaque individu i, mettre à jour la matrice  $(\operatorname{diag}(Q) = Q(Q_{xx}, Q_{yy}, Q_{zz}))$  et appliquer le KF sur toute la base de données en calculant :  $C_i = \sum_{j=0}^N C_j^2$  où N est le nombre de pas de données stockées dans la base.
- 3. Réaliser une itération de l'algorithme génétique avec  $C_i$  le poids associé à chaque individu i.

- 4. Reproduire à partir de 1, avec la nouvelle génération d'individus si aucune des deux conditions d'arrêt n'est satisfaite (n itérations ou coût minimum inférieur à une certaine valeur).
- 5. L'individu qui a le plus grand poids contient les éléments cherchés de la matrice Q.

Les solutions obtenues à chaque lancement de l'algorithme ne sont pas identiques. Cependant, ils sont tous acceptables.

#### 4.1.3 Diagnostic de défauts des capteurs de localisation



FIGURE 4.2 – Diagnostic des capteurs de localisation à l'aide de KF

La plupart de robots mobiles évoluant en milieu extérieur possèdent, entre autres, des capteurs de localisation : GPS, centrale inertielle (IMU) et odomètres ou encodeurs des roues. Les informations issues de ces capteurs sont généralement fusionnées pour former ce que l'on appelle "les fournisseurs de position" qui produisent la position instantanée du robot. Cette fusion ne "coûte pas cher" en terme de temps de calcul.

Une des architectures possibles de fusion consiste à générer deux fournisseurs de position (Pose Provider PP). Le premier fournisseur de position est la fusion des données de l'IMU et des odomètres, le deuxième est la fusion des trois capteurs cités précédemment. La méthode proposée à base de KF pour diagnostiquer ces capteurs consiste à appliquer un filtre sur chaque fournisseur de position comme illustré à la figure 4.2.

En surveillant la distance de Mahalanobis et le vecteur de résidus simples de chacun des filtres, la FDI des trois capteurs de localisation se fait tel qu'expliqué au tableau 4.1. En effet, un défaut GPS apparaît uniquement dans le PP1. Les défauts de l'IMU et des odomètres apparaissent dans les deux PP. La différence entre ces deux défauts est que le défaut IMU provoque un grand écart entre l'angle mesuré du robot et celui prédit par le filtre.

Les deux variantes du filtre de Kalman, EKF et UKF, peuvent être appliquées à cette architecture.

Défaut	Distance de Mahalanobis du	Distance de Mahalanobis du	$  heta - \hat{ heta} $
	filtre sur PP1	filtre sur PP2	
GPS		Х	
IMU	Х	Х	Х
Odomètres	Х	Х	

Tableau 4.1 – Table de vérité servant à l'isolement des défauts

## 4.1.4 Diagnostic de défauts laser

Nous proposons dans cette section une méthode se basant sur le KF (EKF ou UKF) pour le diagnostic de perturbations laser. Comme il est illustré à la figure 4.3,  $\alpha_i$  et  $r_i$  sont respectivement l'orientation et la distance de l'objet *i* par rapport au robot.



FIGURE 4.3 – Notation, scan laser

Les étapes principales de cet algorithme sont :

- 1. Segmentation du scan pour trouver tous les objets qui apparaissent dans le scan.
- 2. Vérification de la présence des objets connus.
- 3. Suivi de l'évolution de la position des objets par rapport au robot.
- 4. Détection des défauts par la surveillance des résidus.

Tout d'abord, la segmentation est faite en mesurant la distance entre deux points consécutifs et l'angle  $\beta$  entre les deux droites (robot-objet<sub>i</sub>) et (robot-objet<sub>i+1</sub>) [16]. Dans la figure 4.4, le point B représente l'objet 1 et le point A représente l'objet 2. L'angle  $\beta$  est calculé comme dans l'équation 4.7.



FIGURE 4.4 – Exemple pour calculer  $\beta$ 

$$\beta = \arctan \frac{||BH||}{||HA||} = \arctan \frac{r_2 \sin \alpha}{r_1 - r_2 \cos \alpha}$$
(4.7)

La distance entre les deux objets (les deux points détectés) peut être calculée comme dans l'équation 4.8.

$$r_{12}^2 = r_1^2 + r_2^2 - 2r_1 r_2 \cos\left(\alpha_2 - \alpha_1\right) \tag{4.8}$$

Deux points successifs appartiennent à un même objet si la distance entre ces deux points est inférieure à  $r_{max}$  et si l'angle  $\beta$  correspondant est supérieur à  $\beta_{min}$ . Cette démarche nous permettra de détecter les objets présents dans le scan.

Suite à cette phase de segmentation, nous aurons une liste des objets détectés. En prenant l'hypothèse que des objets connus immobiles sont mis dans des endroits fixes, l'identification des objets qui doivent apparaître dans le scan parmi ceux de la base de données des objets connus se fait en fonction de la position actuelle du robot. Par exemple, si à l'instant t, un objet o se situe juste derrière le robot en dehors de la portée du scan, o ne doit pas apparaître dans le scan à cet instant. Ensuite, une comparaison se fait entre les objets détectés dans le scan et les objets qui doivent apparaître. Cette phase permet de vérifier si le laser est fonctionnel ou pas.

Ensuite, un EKF flexible suit les objets détectés. À chaque objet i on associe trois variables d'états : la projection du vecteur reliant le robot à l'objet sur l'axe x notée  $r_x$ , sa projection sur l'axe y notée  $r_y$  et l'orientation de l'objet par rapport au robot notée  $\alpha$ . Lorsque l'objet n'est pas détecté, ces valeurs associées dans l'EKF (dans le vecteur d'état, la matrice de transition et la jacobienne) sont nulles. Le modèle non linéaire implémenté dans l'EKF dans le cas d'un robot de type skid-steering est décrit dans l'équation 4.9.

$$r_{x}^{k+1} = r_{x}^{k} + v \cos \theta dt$$

$$r_{y}^{k+1} = r_{y}^{k} + v \sin \theta dt$$

$$\alpha^{k+1} = \alpha^{k} + \frac{\pi}{2} - (\arctan \frac{r_{x}^{k+1}}{r_{y}^{k+1}} + \arctan \frac{r_{x}^{k}}{r_{y}^{k}}) + w dt$$
(4.9)

x, y et  $\theta$  sont la pose du robot. v et w sont les commandes en vitesse linéaire et en vitesse angulaire.

Finalement, la surveillance de la distance de Mahalanobis et du vecteur de résidus simples de l'EKF aboutissent à la détection des perturbations laser.

# 4.2 Les Relations de redondance

Le diagnostic basé sur l'analyse de relations de parité est une méthode naturelle de diagnostic qui consiste à vérifier la consistance entre les différentes variables d'un système. Elle est basée sur la construction de variables à partir de deux sources a minima (modèles, capteurs, entrées et sorties). Une comparaison de celles-ci conduit à la génération de résidus utilisés ensuite pour la détection d'un défaut (voire l'isolement selon les cas). Par exemple, la vitesse linéaire du robot, peut être calculée à l'aide de données provenant des encodeurs, du GPS  $(\frac{\Delta(x)}{\Delta(t)})$  ou bien de la commande et du modèle du système. Ces relations varient en fonction du type de robot et des capteurs dont il est équipé. Nous présentons l'application de cette méthode dans le chapitre 5.

# 4.3 L'estimation récursive de paramètres

Lorsqu'un modèle a une ou plusieurs variables inconnues, les outils d'estimation de paramètres nous permettent de les identifier à partir d'une base de données représentative [72]. En supposant des défauts additifs et multiplicatifs uniquement, la représentation est montrée dans l'équation 2.1. Nous pouvons estimer les  $\beta$  correspondants à chaque mode de fonctionnement (normal ou défaut particulier). Ceci est réalisable en appliquant l'un des algorithmes d'estimation sur une base de données, une fois pour chaque mode. Par contre, l'estimation d'un paramètre exige l'existence d'une architecture valable de ce paramètre, comme celle de l'équation ci-dessus par exemple.

Cette opération se fait hors ligne dans un premier temps. Les  $\beta$  de chaque mode connus, ils sont ensuite utilisés pour le paramétrage de certaines méthodes de diagnostic comme, par exemple, l'estimation récursive des paramètres des défauts ou encore l'estimation multi-modèles.

L'estimation récursive de paramètres [39] consiste à estimer en temps réels les  $\beta$  instantanés, dont l'ajustement itératif est décrit à l'équation 4.10 :

$$\hat{\beta} = \hat{\beta}(t) + K(t)(y(kt) - \hat{y}(t)) \tag{4.10}$$

K, y et  $\hat{y}$  sont respectivement le gain instantané, la sortie mesurée et la sortie prédite avec le modèle. L'état actuel du système est celui pour lequel les paramètres correspondent aux plus proches de ceux estimés en temps réel :  $\hat{\beta}(t)$ .

De nombreux algorithmes d'estimation de paramètres de la littérature peuvent être utilisés pour la FDI. L'estimation par les moindres carrés (Least Square Estimation LSE) est largement utilisée car elle est générale et facile à implémenter [83] [40].

L'objectif de base de la LSE est d'ajuster les paramètres de la fonction du modèle pour s'adapter au mieux à un ensemble de données. Supposons que nous avons n points  $(x_i, y_i)$ , avec i = 1...n, où  $x_i$  est l'entrée et  $y_i$  est la sortie qui peut être trouvée par observation. La fonction du modèle a la forme  $f(x, \beta)$ , où  $\beta$  est un vecteur de m paramètres à estimer. Le but est de trouver les bêtas qui minimisent la somme C de résidus au carré, autrement appelée fonction de coût :

$$C = \sum_{i=1}^{n} (y_i - f(x_i, \beta))^2$$
(4.11)

Si le modèle de régression est linéaire, la fonction du modèle est une combinaison linéaire des paramètres  $f(x,\beta) = \sum_{i=1}^{m} \beta_j g(x)$  où g(x) est une fonction connue de x. Dans ce cas, l'estimation des moindres carrés est donnée par :

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{4.12}$$

Ainsi, les paramètres de défauts ( $\beta_{fault}$ ) de chaque mode peuvent être estimés hors ligne à l'aide de cette méthode grâce à une base de données suffisante contenant des valeurs d'entrée et de sortie (par exemple, commande et position) pour chaque mode (robot normal ou défectueux).

Connaissant les paramètres de défauts, on peut utiliser l'estimation récursive de paramètres (recursive parameter estimation : RLS) pour calculer les valeurs des paramètres réels. Une comparaison avec des valeurs pré-estimées conduit à déterminer le mode du robot.

Plutôt que de recalculer la LSE à partir de zéro à chaque pas de temps, l'algorithme RLS, à l'instant k, met à jour la valeur estimée actuelle en tenant compte des nouvelles données à chaque

pas de temps. Ayant  $\hat{\beta}_{(N)}$ , en utilisant le nouvel échantillon de données  $(x_{N+1}, y_{N+1})$ ,  $\hat{\beta}_{(N+1)}$  est une mise à jour de  $\hat{\beta}_{(N)}$ . La forme simple de l'équation de mise à jour peut être écrite ainsi :

$$\hat{\beta}_{(N+1)} = \hat{\beta}_{(N)} + \frac{1}{\mu} (X_{(N)}^T X_{(N)})^{-1} x_{N+1}^T (y_{N+1} - x_{N+1} \hat{\beta}_{(N)})$$
(4.13)

où  $\mu < 1$  le facteur d'oubli. Plus le  $\mu$  est petit, plus l'importance accordée aux dernières mesures par rapport aux plus anciennes est élevée.



FIGURE 4.5 – FDI par l'estimation récursive de paramètres

En raison des bruits et des imperfections du modèle, l'estimation des paramètres peut ne pas être très précise. Prendre une décision en regardant uniquement les distances entre les valeurs estimées et les valeurs pré-estimées pour chaque mode est donc risqué. Pour remédier à ce problème, l'analyse discriminante linéaire (Linear Discriminant Analysis LDA) est largement utilisée dans la classification [80]. C'est un algorithme simple et efficace. L'idée de base est de projeter les données dans un nouvel espace afin de maximiser la séparabilité des classes, c'està-dire de maximiser la distance entre les éléments appartenant aux différentes classes et de minimiser la propagation au sein de la classe elle-même. Évidemment, pour obtenir ces classes, une étape d'apprentissage supervisé est requise. La figure 4.5 récapitule cette méthode. Cependant, un compromis doit être fait entre robustesse et délai de détection. Selon le facteur

Cependant, un compromis doit être fait entre robustesse et delai de detection. Selon le facteur d'oubli (équation (4.13)), l'algorithme peut être plus ou moins sensible aux changements de mode (transitions).

# 4.4 Multiple Model Adaptive Estimation (MMAE)

Après avoir estimé les paramètres des défauts, une intégration de ces paramètres dans le modèle du robot (équation 4.10) fournit plusieurs modèles du système (un modèle pour chaque défaut). Le diagnostic se fait alors en surveillant, la consistance des mesures avec les sorties de l'ensemble de ces modèles. L'état actuel du système est celui correspondant au modèle le plus consistant avec les mesures.



FIGURE 4.6 – Estimation adaptative multi-modèles

La figure 4.6 illustre l'estimation adaptative multi-modèle où n+1 modèles sont implémentés avec des filtres de Kalman; méthode permettant ainsi d'isoler n défauts. L'un de ces modèles représente le système normal, les autres le modèle avec un défaut spécifique. Le module de décision analyse le résidu et la matrice de covariance de chaque filtre et produit une probabilité  $P_n$  proportionnelle à la consistance de sa sortie avec la mesure. Une somme des estimations pondérées par les probabilités relatives donne l'estimation finale qui sert pour le contrôle tolérant aux défauts.



FIGURE 4.7 – FDI par MMAE

Dans la méthode proposée à base de MMAE, le filtre utilisé est l'UKF. Comme présenté à la figure 4.7, les résidus de chaque filtre sont bloqués dans une fenêtre glissante de taille m. Ensuite, un

filtre passe-bas de Butterworth est appliqué sur la somme de chaque vecteur de résidus bloqués. Ceci permet de réduire les fausses alarmes dues au bruit. Enfin, la comparaison de ces sommes conduit à l'identification du mode réel. Le filtre ayant le résidu minimum correspondra au mode actuel. Encore une fois, en fonction de la taille de la fenêtre m, un compromis entre le délai et la robustesse de la détection doit être envisagé.

# 4.5 Filtres à particules (PF)

### 4.5.1 PF Standard

Le filtre à particules (PF) [77] est une méthode de Monte-Carlo séquentielle. Contrairement au filtre de Kalman, aucune hypothèse n'est prise ni sur la linéarité ni sur le type du bruit. Il traite le système comme un ensemble de variables discrètes ou modes (normal, défaut1, défaut2...), et de variables continues observables ou non (vitesses, accélérations, orientations, ...). Donc, théoriquement, il peut être appliqué sur n'importe quel système. Son algorithme pour un système hybride à variables continues (vitesse, angle, ...) et à états discrets (normal, défaut1, défaut2, ...) est le suivant :

- 1. Pour N particules  $p^i, i = 1, ..., N$ , échantillonner les modes discrets  $d_0^i$  reposant sur la distribution connue  $P(D_0)$ .
- 2. Échantillonner les variables continues  $c_0^i$  pour chaque particule  $p^i$  reposant sur la distribution connue  $P(C_0, d_0^i)$ .
- 3. Pour chaque pas de temps :
  - (a) Pour chaque particule  $p_t^i = (D_t^i, C_t^i)$ :
    - i. Rééchantillonner le mode discret :  $d_t^i \sim P(D_t \setminus d_{t-1}^i)$
    - ii. Rééchantillonner les paramètres continus :  $c_t^i \sim P(C_t \backslash d_t^i, c_{t-1}^i)$
    - iii. Calculer le poids de la particule :  $w_t^i \sim P(o_t \backslash d_t^i, c_t^i)$
  - (b) Rééchantillonner N nouveaux échantillons  $p_{t+1}^i$  en respectant les poids calculés :

Avec  $D = d_1, ..., d_k$  l'ensemble des modes discrets dans lesquels le système peut être trouvé,  $C = c_1, ..., c_l$  l'ensemble des variables d'état continues qui captent l'évolution dynamique du système et O l'ensemble des variables observables.

Le filtre à particules possède un certain nombre de propriétés qui en font un algorithme d'approximation intéressant :

- Il peut être utilisé pour des systèmes hybrides, non linéaires et avec des distributions non gaussiennes;
- Il peut être facilement ajusté à la puissance de calcul disponible par un changement du nombre de particules. Plus le nombre de particules est important, plus le PF est précis. Enfin, le temps de calcul d'un filtre à particules dépend uniquement du nombre de particules, et non de la complexité du modèle;
- Il est simple et facile à implémenter.

Cependant, le PF n'est pas idéal. Cette méthode présente des inconvénients :

- En cas de faibles capacités de calcul, le nombre de particules est faible; ce qui affecte la précision.
- En raison des faibles probabilités d'apparition des modes de défaut, les défauts peuvent ne pas être détectés.
- Le nombre de particules nécessaires augmente exponentiellement avec la dimension de l'espace d'états continus.

En raison de ces inconvénients, le PF n'est pas parfaitement adapté au diagnostic en robotique mobile. Différentes variantes de cet algorithme ont été proposé. Nous les présentons avant d'expliquer la version modifiée que nous proposons.

## 4.5.2 Filtre à particules sensible aux risques

Pour surmonter le problème de faible probabilité d'occurrence des défauts, le filtre à particules sensible aux risques (Risk Sensitive Particle Filter (RSPF)) [77] introduit le concept de coût lors du calcul des poids. En effet, le RSPF peut être utilisé pour surveiller les événements à faible probabilité.

Cette méthode introduit deux modifications à l'algorithme d'un PF standard (paragraphe 4.5.1):

- ligne 1 : Échantillonner le mode discret  $d_0^i$ , en tenant compte du produit  $\gamma_0 r(d_0) P(D_0)$ . Avec  $r(d_t)$  la fonction de coût et  $\gamma_t$  une constante de normalisation qui assure que l'équation est bien une distribution de probabilité.
- ligne 3.a.iii : le poids est maintenant calculé de cette façon :  $w_t^{(i)} = \frac{r(\hat{c}_t^{(i)})}{r(\hat{c}_{t-1}^{(i)})} P(o_t \setminus \hat{d}_t^{(i)}, \hat{c}_t^{(i)})$

Toutefois, le RSPF ne résout pas tous les types de problème. La performance du RSPF se dégrade lorsqu'il existe de nombreux états de défaut similaires à haut risque. De plus, sa performance dépend fortement de la fonction de coût, qui n'est pas forcément facile à définir.

## 4.5.3 Filtre à particules à résolution variable

Les systèmes qui comprennent un grand espace d'état exigent un nombre énorme de particules pour que chaque état soit bien représenté. Le Filtre à particules à résolution variable (Variable Resolution Particle Filter (VRPF)) [77] introduit la notion de particules abstraites. Les particules peuvent représenter des états individuels ou des ensembles d'états. Une seule particule abstraite peut suivre simultanément plusieurs états; ce qui permet de diminuer le nombre de particules. Les états avec des modes de transition et d'observation semblables sont alors agrégés ensemble. Cependant, le VRPF a certaines limites. Il est avantageux uniquement dans le cas où certains modes se ressemblent c.à.d qu'ils possèdent des variables similaires.

## 4.5.4 Filtre à particules Rao-Blackwell

L'un des inconvénients des PF est que le nombre de particules nécessaires augmente exponentiellement avec la dimension de l'espace d'états continus. Le filtre à particules Rao-Blackwell(RBPF) [22] échantillonne uniquement les modes discrets. Cette méthode utilise ensuite un KF permettant d'estimer la moyenne et la variance  $(\mu_t^{(i)}, \Sigma_t^{(i)})$  des variables continues au lieu de les échantillonner. L'avantage d'un RBPF par rapport à un PF standard est que le nombre de particules nécessaires pour obtenir une bonne estimation est moins élevé. Une étape supplémentaire de "look-ahead" peut être effectuée à l'aide d'un KF. Cette étape permet de prédire l'état futur, ce qui augmente les performances. L'algorithme s'appelle alors RBPF2 ou "look-ahead" RBPF. Cependant, les RBPF utilisant un ou plusieurs KF, en ont la même limitation, c'est-à-dire que la méthode ne peut être utilisée que pour des systèmes linéaires avec des mesures gaussiennes, des distributions de bruit de processus gaussiennes et des modèles connus pour les paramètres continus.

## 4.5.5 Filtre à particules Gaussien et Filtre à particules Unscented

Le filtre à particules gaussien GPF [38] est très similaire au RBPF, sauf qu'il traite des modèles non linéaires à l'aide d'un UKF, ceci afin d'estimer les variables non linéaires. La version Unscented (UPF) est très proche d'un GPF. En complément de ce dernier, elle introduit une étape d'échantillonnage de la sortie de l'UKF mis en œuvre. Deux extensions de ces algorithmes, appelés GPF2 et UPF2 [38], consistent à ajouter une étape de prédiction des variables continues à l'aide d'un UKF. Cette étape permet d'améliorer la précision de l'estimation, mais demande un temps de calcul plus important.

Théoriquement, ces deux versions de filtre à particules sont performantes lorsqu'elles sont appliquées sur des systèmes non linéaires. Une évaluation par simulation des performances d'un PF ordinaire, d'un GPF2 et d'un UPF est réalisée dans [12]. Le diagnostic portait sur un robot à six roues modélisé par six variables continues. Le résultat attendu du diagnostic est l'état du robot (identification de l'état du robot parmi 6 états discrets). Les résultats montrent que la performance de diagnostic de l'état du système est meilleure avec le GPF2 et l'UPF que celle obtenue avec le PF standard. Le GPF2 donne un taux de bon diagnostic de 75%, l'UPF de 35% et le PF de 20% seulement pour un temps de calcul de 0.8s. Le GPF2 donne un taux de bon diagnostic de 90%, l'UPF de 90% et le PF de 40% pour un temps de calcul de 8s. Ces résultats montrent notamment la relation directe entre la performance d'un filtre à particules et le temps de calcul.

Les auteurs de [38] concluent sur le fait que l'UPF donnerait de meilleurs résultats lorsque l'état du système suit une distribution de probabilité très différente d'une distribution gaussienne tandis que le GPF2 le surpasse lorsque la covariance de cette distribution est grande. Dans ce cas l'échantillonnage introduit une variance élevée sur l'estimation des états.

Plus généralement, il est important de noter que tous ces algorithmes partagent les mêmes limitations que l'UKF.

## 4.5.6 Proposition d'un filtre à particules : Filtre à Particules Mixé

D'un côté, la version standard du PF n'exige pas forcément un modèle connu, mais elle souffre des inconvénients expliqués dans le paragraphe 4.5.1, notamment la grande puissance de calcul nécessaire. De l'autre côté, la version gaussienne possède des avantages par rapport à la version standard, mais elle exige un modèle connu des variables continus concernées. La version proposée considère deux types de variables continues : 1) à modèle d'évolution connu et 2) à modèle d'évolution inconnu. Cette version a été conçue principalement pour améliorer la performance des deux méthodes MMAE et l'estimation récursive de paramètres en les fusionnant. Dans ce paragraphe, l'algorithme du Filtre à Particules Mixé (PFM) pour fusionner ces deux méthodes est expliqué. Pour résumer, il existe deux algorithmes qui nécessitent un compromis entre le délai de détection et la robustesse. Notre proposition est d'utiliser une version modifiée de PF pour fusionner ces deux algorithmes, de manière à tirer le maximum des avantages de chacun.

// Initialization foreach particle  $p^i$  of N particles do // Sample discrete modes uniformly  $d^i \sim \mathcal{U}(0, maxmode)$  // Sample  $B^i$  with respect to  $d^i$  $\beta_{ii}^{i} = \beta_{ii}^{d^{i}} + \mathcal{N}(0, \sigma_{ii}) \quad (x^{i}, y^{i}, \theta^{i}) \leftarrow (x, y, \theta) \text{ from } UKF_{d^{i}}$ end foreach Time step do foreach particle  $p^i$  of N particles do /\* Resample  $B^i$  with respect to  $\Delta B$  and recompute discrete modes \*/ if  $|\Delta B| > threshold$  then angle =  $(\overrightarrow{\Delta B}, B^i B)$  if  $\cos(angle) \stackrel{\circ}{\phantom{a}} \theta$  then  $| dist = |B^iB| * \cos(angle) + k_{push} * \Delta B$ else  $dist = k_{push} * \Delta B$ end  $B^i + = \mathcal{U}(dist/2, dist)$ else // Converge  $B^i$  toward B $B^i + = \mathcal{U}(k_{converge}/2, k_{converge})$ end  $d^i$  $LDA(B^i)$ /\* Push betas toward the center of their mode to avoid  $\leftarrow$ divergence, and read UKF values \*/  $B^{i} + = \mathcal{N}(0, \sigma) * sign(B - B^{i}) \quad (x^{i}, y^{i}, \theta^{i}) \leftarrow (x, y, \theta) \text{ from } UKF_{d^{i}}$ end foreach particle  $p^i$  of N particles do /\* Compute wheights \*/  $w^i = k_1 * \mathcal{N}(k_0 * (B^i - B)) + k_2 * \mathcal{N}(r_{ukf_{di}})$  // Normalize wheights  $w^i = \frac{w^i}{\sum_{i=0}^N w^i}$  /\* Resample N new particles using resampling [43] \*/  $P_{t+1} \leftarrow resampling(P_t)$ end  $State = argmax(d^i)$ end

#### ALG 4.1 – L'algorithme PFM proposé

Chaque particule contient des variables prédites par l'UKF : le vecteur de l'estimation des paramètres de défaut  $B = \beta_1, ..., \beta_n$  et l'estimation en mode discret d. Supposons que les valeurs prédites par l'UKF soient la position et l'orientation, à savoir la pose  $(x, y, \theta)$ . Comme le montre la figure 4.8, pour une particule ayant le mode i, la pose  $(x, y, \theta)$  correspondra à l'UKF implémentant le même mode i.  $B_{resampled}$  sera rééchantillonné à l'intérieur de la particule et comparé à B calculé par l'algorithme RLS pour générer un résidu. Ce résidu et celui du filtre UKF correspondant sont normalisés avec une distribution gaussienne centrée. La somme de ces deux résidus va générer le poids des particules.  $k_0, k_1 < 1$  et  $k_2 < 1$  sont des gains positifs permettant de mettre à l'échelle les deux résidus et ainsi privilégier l'algorithme le plus robuste.



FIGURE 4.8 – Une particule du PFM

L'algorithme de ré-échantillonnage prend en compte la variation de B (obtenue de RLS) et attire  $B^i$  de toutes les particules dans sa direction. Ensuite, la LDA est appliquée pour calculer le mode de particules en fonction de nouvelles valeurs B. L'algorithme 4.1 montre un pseudo-code de cette méthode.

# 4.6 L'apprentissage profond

Comme évoqué dans le paragraphe 2.6, l'apprentissage profond (Deep-Learning (DL)) est un algorithme basé sur l'intelligence artificielle. Sa conception consiste à définir :

- 1. l'architecture du réseau de neurones mis en œuvre, c'est-à-dire, le nombre de couches et de neurones dans chaque couche et le type de connexion entre eux;
- 2. la méthode d'apprentissage, c'est-à-dire la méthode avec laquelle les poids des liaisons entre les différents neurones sont calculés.

Il s'agit d'un réseau de neurones entrainé en deux phases : un apprentissage non supervisé (une base de données qui ne précise pas les sorties correspondant aux entrées) suivi par un apprentissage supervisé. Les quatre architectures principales de l'apprentissage profond sont [53] :

— La machine de Boltzmann restreinte. Une machine de Boltzmann est formée d'un graphe bipartie (Figure 4.9) de deux couches de neurones. L'une reçoit l'entrée et l'autre est caché. Les deux couches sont reliées par une connexion bidirectionnelle et les neurones de la même couche sont indépendants entre eux. L'apprentissage non supervisé se fait avec l'échantillonnage de Gibbs [24] (méthode d'échantillonnage par chaînes de Markov), puis les poids finaux sont obtenus en maximisant une fonction de vraisemblance.



FIGURE 4.9 – Structure d'une machine de Boltzmann restreinte

— Le réseau de croyance profond ou "Deep Belief Network" (DBN). Ce réseau peut être vu comme un ensemble de machines de Boltzmann mises en cascade. Formées de trois couches cachées et d'une couche visible comme le montre la Figure 4.10, les connexions entre les différentes couches adjacentes sont complètes. Elles sont dirigées de haut en bas, sauf celles entre les deux couches les plus hautes (couches cachées 2 et 3) qui sont bidirectionnelles. L'apprentissage non supervisé de ce réseau se fait en ajustant les poids couche par couche de bas en haut, puis un algorithme d'apprentissage supervisé est utilisé dans la direction inverse, de haut en bas.



FIGURE 4.10 – Structure d'un DBN

— L'auto encodeur (AE). l'AE une couche cachée de type "Feed-forward". Il vise à reconstruire l'entrée avec d'autres formes au lieu de prédire la sortie. L'AE peut être utilisé pour la réduction de taille de paramètres, pour la compression de données mais aussi en classification où il peut servir en prétraitement effectif des données ou bien directement pour de la classification. Sa structure est divisée en deux blocs comme le montre la figure 4.11. Durant le codage, l'AE convertit l'entrée x en une représentation cachée h à l'aide d'une matrice de poids w. Le décodage reconstruit le format original des données pour obtenir x à l'aide de w' qui est théoriquement la transposée de w. L'apprentissage non supervisé se fait à l'aide de la propagation Feed-forward pour chaque entrée. La déviation est mesurée à l'aide de l'erreur quadratique; les poids étant obtenus d'après la propagation de cette erreur. Ensuite, l'algorithme de descente de gradient est utilisé pour ajuster ces poids avec une méthode d'apprentissage supervisé standard (retro propagation).



FIGURE 4.11 – Structure d'un auto-encodeur

— Le réseau de neurones à convolution ou "Convolutional neural networks" (CNN). Très performant en traitement de données bidimensionnelles (images, vidéos...), le CNN est un réseau de neurones multicouches qui contient deux types de couches : les couches à convolution (C-couches) et les couches de sous-échantillonnage (S-couches) connectées alternativement. Prenons l'exemple de la Figure 4.13. Dans les C-couches, n convolutions de l'entrée se font avec n filtres  $f_x$  puis un biais  $b_x$  s'y ajoute.  $f_x$  et bx sont déterminés suite à l'entraînement du CNN, permettant de générer la couche de convolution  $C_x$ . Le sous-échantillonnage consiste à additionner chaque 4 pixels voisins, les pondérer avec  $w_{x+1}$ , y ajouter un biais entrainable  $b_{x+1}$  puis l'entrer dans une fonction sigmoïde pour produire enfin une carte  $S_{x+1}$  de taille réduite par rapport à  $C_x$  [13]. Ce traitement de données se reproduit pour chaque paire C-couche S-couche, avant d'être réarrangés (pixelisés) puis rentrés dans un réseau de neurones conventionnel. L'apprentissage d'un CNN est similaire à celui d'un réseau de neurones standard (retro propagation). La forme de base des neurones construisant les modèles est présentée à la figure 4.12, avec  $x_i$  le vecteur d'entrée et  $w_i$  les poids à ajuster.



FIGURE 4.12 – Forme de base d'un neurone



FIGURE 4.13 – Structure d'un CNN

Une variante importante de neurone est le Long-Short Term Memory (LSTM). LSTM est un

neurone de type récurrent [67]. Il est composé de cellules, chaque cellule a un état associé. Il est conçu principalement pour traiter les informations séquentielles. Il accumule des informations du passé pour prédire l'état actuel.

Pour contrôler l'ajout d'informations provenant de l'état de la cellule, celle-ci contient des portes. Une porte est une couche de réseau de neurones suivie d'une multiplication ponctuelle. La fonction d'activation de la porte est généralement sigmoïde, avec des valeurs comprises entre 0 et 1.

## 4.6.1 Proposition de modèles basés sur l'apprentissage profond

Les méthodes proposées basées sur l'apprentissage profond consistent à traiter les résidus pour déterminer l'état réel (ou le mode) du robot. Les résidus sont obtenus par des équations de redondance analytique comme expliqué dans 4.2.

Deux modèles sont proposés. Le premier est un modèle de classificateur avec couches cachées(comme celui du DBN). Le deuxième modèle est composé d'une étape de codage d'Autoencoder suivie d'un classificateur. Les deux modèles contiennent une couche d'entrée LSTM pour prendre en compte l'aspect temporel.

L'application de cette technique consiste à :

- 1. Recueillir les données nécessaires pour l'entraînement, couvrant tous les modes et les commandes possibles (toutes les combinaisons commande/mode);
- 2. Entraîner le modèle hors ligne;
- 3. Exécuter en ligne sur le vecteur d'entrée provenant de l'application des relations de redondance sur les données capteurs.

L'ajustement des paramètres nécessaires se fait de façon empirique, i.e. les fonctions d'activation, le nombre de couches et les autres paramètres d'apprentissage.

Dans notre application, les caractéristiques des modèles proposés sont ajustés comme décrit à la figure 4.14.

Les fonctions d'activation adoptées dans ces deux schémas sont sigmoïde, relu et softmax. Nous expliquons brièvement ces fonctions ci-dessous.



FIGURE 4.14 – Modèles proposés basés sur l'apprentissage profond

Sigmoïde : La forme de cette fonction est :

$$\Phi(z) = \frac{1}{1 + e^{-z}} \tag{4.14}$$

Cette fonction est bornée entre 0 et 1. Son graphe prend la forme d'une lettre s (figure 4.15)



FIGURE 4.15 – Graphe d'un sigmoïde

On peut remarquer que lorsque z est proche de 0, la variation entre les valeurs de  $\Phi(z)$  est importante. Cette fonction a tendance à attirer les valeurs Y à l'une de ces deux extrémités engendrant une meilleure classification entre les valeurs proches.

**Relu** : Rectified Linear Unit est la fonction d'activation la plus utilisée dans le monde à l'heure actuelle. Elle est utilisée dans presque tous les réseaux de neurones convolutionnels ou d'apprentissage profond. L'équation 4.15 la décrit.

$$R(z) = \max(0, z) \tag{4.15}$$

La figure 4.16 représente le graphe de cette fonction. Cette fonction varie sur l'intervalle  $[0, +\infty[$ . Imaginons le cas d'un réseau avec des pondérations aléatoires initialisées (ou normalisées). Près de 50% du réseau génère 0 en raison de la caractéristique de ReLu (sortie 0 pour les valeurs négatives d'entrée). Cela signifie que moins de neurones sont activés (activation clairsemée) et que le réseau est plus léger. L'entraînement d'un tel réseau est plus rapide, et son application demande moins de puissance.
60 Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance



FIGURE 4.16 – Graphe de la fonction Relu

**Softmax** : c'est une fonction d'activation très intéressante car elle associe la sortie sur une plage de [0,1] dont la somme totale est égale à 1. La sortie de Softmax est donc une distribution de probabilité. Elle est souvent utilisée dans la dernière couche d'un classificateur basé sur un réseau de neurones. Sa fonction mathématique est représentée dans l'équation 4.16, où z est le vecteur d'entrées dans la couche de sortie et j référence les unités (neurones) de sortie de 1, 2, 3..., k:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{pour } j=1,...,K$$
(4.16)

Cette fonction d'activation est alors utile pour les problèmes de classification multiple.

## 4.7 Proposition d'une méthode de diagnostic pour les nœuds communicants basée sur le vecteur de confiance

Pour des applications de suivi ou de localisation dans la robotique, des technologies de communication comme le wifi ou l'UWB peuvent être appliquées. Par conséquent, il est intéressant de rajouter à la MHD une sous méthode permettant le diagnostic des erreurs de communication. D'après la revue [59], parmi les diverses méthodes de diagnostic des nœuds communicants, la méthode "Trust Matrix" a prouvé sa capacité à diagnostiquer le défaut de perte de communication avec un taux de bonne détection supérieur à 80%. Cette méthode consiste à ajuster les valeurs d'une matrice représentant le taux de confiance accordé à la communication entre les paires de nœuds selon la réussite ou l'échec d'une tentative de communication entre ces deux nœuds.

Une approche inspirée de cette méthode est conçu pour détecter et isoler ce défaut suite à la perte de communication entre des nœuds. Une communication entre des tags UWB est considérée. Un tag, appelé initiateur, ouvre une communication avec un autre, appelé récepteur. À l'instant t, un seul initiateur discute avec un seul récepteur. En prenant ces hypothèses, et en supposant que n initiateurs discutent avec m récepteurs, l'algorithme développé est comme suit :

1. Construire la matrice instantanée d'état de communication M. Les lignes représentent les initiateur I et les colonnes représentent les récepteurs R.  $M_{ij} = 1$  si la tentative de communication entre  $I_i$  et  $R_j$  (I et  $R \in T$ ) est réussie.  $M_{ij} = 0$  sinon.

$$M = \begin{array}{cccc} & r_1 & r_2 & \dots & r_m \\ i_1 & m_{11} & m_{12} & \dots & m_{1m} \\ m_{21} & m_{22} & \dots & m_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ i_n & m_{n1} & m_{n2} & \dots & m_{nm} \end{array}$$
(4.17)

- 2. Les lignes et les colonnes qui ne contiennent que des zéros correspondent à des tags défectueux. Ces tags sont identifiés et un coefficient réduit (de l'ordre de 0.2) leur est attribué. Le coefficient des autres tags est maintenu à 1. Ce coefficient (coef) servira à réduire l'effet de ces tags sur le taux de confiance des autres tags.
- 3. Pour chaque ligne et chaque colonne, le taux préliminaire de confiance de chaque tag est calculé comme la somme sur la ligne (pour les initiateurs) ou sur la colonne (pour les récepteurs) :

$$t_i = \frac{\sum \text{état.coef}}{\sum coef}; t_i \in [0, 1]$$
(4.18)

4. Ajuster le taux de confiance des tags en ajoutant ou en retranchant un pas :

$$\operatorname{trust}_{t_i}^t + = \operatorname{signe}(\operatorname{trust}_{t_i}^{t-1} - t_i) * \operatorname{pas}$$
(4.19)

- 5. Analyser les valeurs des taux pour isoler les cas suivants :
  - i Perte fréquente de communication sur un ou plusieurs tags si  $trust_{t_i} < seuil dur$ ;
  - ii Perte moins fréquente de communication sur un ou plusieurs tags si  $trust_{t_i} < seuil soft$ ;
  - iii Communication impossible entre une paire de tags si la perte persiste uniquement entre les deux tags concernés.

## 4.8 Conclusion

Dans ce chapitre, les contributions faites pour adapter les méthodes de diagnostic aux défauts de la robotique mobile sont montrées. Des méthodes à base de KF sont proposées pour isoler les défauts capteurs de localisation et pour le défaut laser. Un algorithme d'auto-paramétrage du KF dans le but d'améliorer la détection des perturbations est présenté. Différentes approches basées sur l'estimation de paramètres des défauts sont proposées. Un algorithme (PFM) fusionnant deux versions (GPF et PF standard) est développé pour le diagnostic des défauts à modèle connu. Ensuite, deux modèles à base d'apprentissage profond sont détaillés. Ils prennent comme entrée un ensemble de résidus issus de la redondance analytique pour prédire l'état du robot. Finalement, un algorithme pour l'isolement des défauts de communication entre des nœuds communicants (UWB) est proposé. Le chapitre suivant montre les résultats de mise en œuvre de ces approches ainsi que les résultats obtenus en appliquant une forme déduite de la MHD sur chacun des défauts considérés.

# Chapitre 5

# Mise en application de la démarche de diagnostic proposée

La mise en œuvre de la démarche de diagnostic proposée est présenté dans ce chapitre ainsi que les résultats obtenus. Pour une présentation claire, ce chapitre est structuré conformément à la figure 3.1. L'application est faite en simulation et sur le robot réel pour les défauts capteurs de localisation et uniquement en simulation pour les autres défauts pour des raisons de sécurité. Sauf contre-indication, l'axe des abscisses dans les figures de résultats représente les itérations.

## 5.1 Définition du contexte d'application de la démarche

## 5.1.1 Environnement d'évaluation

La simulation est développée sous la version Kinetic de ROS (Robot Opeating System) [31], qui est un ensemble de librairies, de logiciels et d'outils open source pour le développement d'applications pour la robotique. Il comporte notamment un simulateur 3D appelé "Gazebo". Il permet de simuler différents robots dans un environnement virtuel et d'avoir accès à de nombreux paramètres entrant dans la gestion de la dynamique de ces derniers. Ainsi, des défauts comme la dégradation d'une roue, le blocage d'une roue, la panne d'un moteur, ou encore des défauts liés à l'environnement comme la présence d'obstacles ou de zones glissantes peuvent être simulés.



FIGURE 5.1 – Exemple d'un environnement sous Gazebo

La figure 5.1 montre un exemple de l'environnement Gazebo. Elle représente un robot de type unicycle (skid-steering) sur un sol plan, entouré de plusieurs objets. La partie en rouge est une

zone de glissement. Celle en vert contient deux obstacles négatifs (trous).

Ce type de simulation est maintenant très utilisé dans le monde de la recherche. Dans [28] par exemple, une simulation des scénarii logistique de coopération de plusieurs robots a été faite sous Gazebo.

ROS intègre une architecture de communication spécifique. Elle assure la communication entre différents processus (nœuds) par l'intermédiaire de voies de communication (topics). Les nœuds qui souhaitent fournir des données avertissent le "maître" de ROS pour publier sur un topic. De l'autre côté, les nœuds qui souhaitent recevoir ces données avertissent le maître qu'ils veulent écouter ce topic. Cette configuration permet d'intercepter les données capteurs et de les modifier avec des programmes tiers pour injecter dans la simulation des défauts capteurs. Ces programmes écoutent les topics provenant des capteurs, les modifient pour y injecter des défauts et les renvoient de nouveau, comme le montre la figure 5.2.



FIGURE 5.2 – Injection de défauts sur les mesures capteurs

Nous pouvons injecter aux données des défauts de différents types :

- Des perturbations (par l'intermédiaire de signaux aléatoires);
- Des biais (en ajoutant une valeur constante);
- Des dérives (en ajoutant une valeur croissante);
- Des blocages (en fixant une valeur).

**Premier scénario** Un premier scénario de test est défini pour tester les réponses des algorithmes aux différents défauts sauf au cas de la communication UWB et du laser, où des conditions particulières sont appliquées. Le signal de commande est tracé à la figure 5.3. En bleu, la commande de vitesse linéaire et en vert, la commande de vitesse angulaire. Il est composé de différentes combinaisons de commandes. Les rectangles noirs séparent les combinaisons de commandes. Dans la partie 1, w = 0 et v bascule entre des valeurs positives et des valeurs négatives successives, c'est-à-dire que le robot avance et recule en ligne droite. Dans la partie 2, v = 0 et w bascule entre des valeurs positives et des valeurs négatives successives, c'est-à-dire que le robot tourne à droite et à gauche sur place. dans la partie 3, v et w basculent entre des valeurs différentes, le robot réalise ici différentes formes de trajectoires circulaires. Enfin, dans la partie 4, le robot effectue une trajectoire sinusoïdale.



FIGURE 5.3 – Le signal de commande du scenario 1

Cette même succession de commandes a été rejouée pour chaque mode (normal, défauts) afin d'étudier le comportement des méthodes de diagnostic sous les différentes conditions. Un défaut est injecté toutes les 15 secondes et dure 15 secondes avant que le mode normal ne soit rétabli. La durée totale du scénario est d'environ 10 minutes.

**Deuxième scénario** Un autre scénario plus concret est construit. Il est présenté à la figure 5.4. Dans ce scénario, le robot se déplace entre quatre étagères. Quand il arrive près d'une étagère, il se déplace lentement, s'arrête une seconde, puis recule et tourne vers l'étagère adjacente. Au cours de sa mission, il doit grimper une rampe (en rouge) et éviter un obstacle (cube blanc) en tournant et en avançant. Il achève cette trajectoire à faible vitesse puis à grande vitesse.



FIGURE 5.4 – Scénario 2

Dans chaque scénario, un défaut est injecté ou retiré tous les 30s. Les résultats sont sauvegardés puis évalués à la fin du scénario.

## 5.1.2 Définition du Robot

Les tests sont faits sur deux robots de test de type Skid-Steering : le "Jaguar" (figure 5.5) et l'Effibot 3. L'axe de rotation des roues est fixé au châssis du robot ; la prise de virage se fait en donnant des vitesses différentes sur les roues de chaque côté du robot.



FIGURE 5.5 - Le robot Jaguar

Ces robots pèsent moins de 20kg. Il ont été conçus respectivement par la société Dr Robot et Effidence, pour des applications indoor et outdoor. Ils possèdent quatre moteurs de 80W; un pour chaque roue sachant que les deux moteurs de chaque côté sont contrôlés par le même contrôleur. Ils peuvent atteindre une vitesse maximale de 3m/s, et peuvent communiquer avec un poste de commande distant via un module Wireless intégré. Ces robots ont été équipés d'un GPS, d'une IMU (centrale inertielle qui donne les accélérations angulaires et linéaires, la vitesse angulaire et l'orientation du robot) et de 4 odomètres (qui donnent la vitesse des roues). Ils peuvent être équipés d'une caméra audio/vidéo, d'un scanner laser qui détecte les obstacles dans son champ de vision (de type Lidar), d'un scanner laser 3D (Velodyne) et d'une caméra 3D (type caméra Kinect).

## 5.1.3 Caractéristiques de l'opérateur

Dans ces travaux l'opérateur est considéré comme expert. Il a alors des connaissances sur les éléments du robot ainsi que sur les signaux envoyés par la MHD (les résidus et les décisions des méthodes de diagnostic). En conséquence, les décisions prises par l'opérateur seront considérées comme les plus fiables.

Cette hypothèse ne sera pas forcément le cas en condition réelle où plusieurs "profils" des opérateurs existent. Dans le cas d'un opérateur non expert, il faudra : 1- simplifier davantage l'interface graphique et 2- considérer que la décision prise par l'opérateur est "discutable" dans le sens où elle ne sera pas forcément la décision la plus fiable.

# 5.2 Mise en œuvre d'OMOPAMED2

## 5.2.1 Identification des défauts

La première étape pour la mise en œuvre d'OMOPAMED2 est de définir les défauts que le système de supervision doit traiter. Ces défauts sont cités au tableau 2.1.

#### 5.2.2 Couples défauts/méthodes

Une première attribution théorique des méthodes de diagnostic aux défauts peut être faite grâce à l'analyse réalisée à la section 2.6.

En bref, les défauts de perturbations capteurs de localisation peuvent être diagnostiqués par les filtres de Kalman, les relations de redondance et l'apprentissage profond. Les défauts de blocage de roue peuvent être diagnostiqués par les méthodes à base de connaissances, l'apprentissage profond et les relations de redondance. Finalement, les défauts laser et UWB peuvent être diagnostiqués respectivement par l'EKF et la méthode basée sur le vecteur de confiance.

#### 5.2.3 Mise en œuvre d'OPAMED2

Cette étape consiste à définir les outils nécessaires pour adapter les méthodes sélectionnées aux défauts auxquels elles sont attribuées et les outils nécessaires pour le conditionnement des données d'entrée des méthodes. La théorie est détaillée dans le chapitre 4, la mise en application des ces outils (paramètres des filtres, ...) est présentée dans la section 5.3 avec l'explication des méthodes associées.

#### 5.2.4 Modèle du robot et des défauts

Le modèle cinématique Newtonien (unicycle) est exprimé dans l'équation 5.1 (avec x, y et  $\theta$  les coordonnées du centre du robot dans le plan 2D - position et orientation) :

$$\begin{aligned} \dot{x} &= v \cos(\theta) \\ \dot{y} &= v \sin(\theta) \\ \dot{\theta} &= w \end{aligned} \tag{5.1}$$

Où v et w sont les commandes de vitesse linéaire et angulaire.

Ensuite, en considérant que  $v_L$  et  $v_R$  sont respectivement la vitesse des roues gauches et celle des roues droites et que (v, w) est le signal de la commande, les équations dans 5.2 expriment la relation entre la commande et la vitesse des roues :

$$v = \frac{v_L + v_R}{2}$$
$$w = \frac{v_R - v_L}{L}$$
(5.2)

Avec L la longueur de l'axe (ou la distance entre les roues droites et gauches). En partant de ces équations et en considérant l'idée que chaque paramètre peut être calculé ou obtenu via différentes méthodes, les équations 5.3 sont tirées en utilisant les notations de la figure 5.6.



FIGURE 5.6 – Représentations des paramètres sur le robot

$$w = \frac{v_r - v_l}{2} \tag{5.3a}$$

$$\dot{\theta} = \frac{v_r - v_l}{L} \tag{5.3b}$$

$$v = \frac{V_r + v_l}{2} \tag{5.3c}$$

$$v_g = \frac{V_r + v_l}{2} \tag{5.3d}$$

$$v_{r1} = v + \frac{Lw}{2} \tag{5.3e}$$

$$v_{r2} = v + \frac{Lw}{2} \tag{5.3f}$$

$$v_{l1} = v - \frac{Lw}{2} \tag{5.3g}$$

$$v_{l2} = v - \frac{Lw}{2} \tag{5.3h}$$

$$v_{r1} = v_g + \frac{L\theta}{2} \tag{5.3i}$$

$$v_{r2} = v_g + \frac{L\theta}{2} \tag{5.3j}$$

$$v_{l1} = v_g - \frac{L\theta}{2} \tag{5.3k}$$

$$v_{l2} = v_g - \frac{L\dot{\theta}}{2} \tag{5.31}$$

$$w = \theta \tag{5.3m}$$

$$v_{r1} = v_{r2} \tag{5.3n}$$

$$v_{t1} = v_{t2} \tag{5.3o}$$

$$v_{l1} = v_{l2}$$
 (5.30)

$$v = v_g \tag{5.3p}$$

Où w et  $\dot{\theta}$  sont des vitesses angulaires en rad/s respectivement données par le signal de commande et mesurées par l'IMU. Les numéros 1 et 2 dans  $v_{x1}$  et  $v_{x2}$  font référence aux roues avants et arrières de chaque côté du robot. v et  $v_g$  sont des vitesses linéaires en m/s respectivement données par le signal de commande et mesurées via GPS grâce à l'équation 5.4.

$$v_g = \frac{\sqrt{(\delta x_g)^2 + (\delta y_g)^2}}{\delta t} \tag{5.4}$$

Où,  $\delta x_g$  et  $\delta y_g$  représentent la distance parcourue par le robot dans le repère 2D absolu XY.

Résidus	IMU	GPS	Commande	Roue <sub>avant gauche</sub>	Roue <sub>arrière gauche</sub>	Roue <sub>avant droite</sub>	Roue <sub>arrière droite</sub>
a			Х	Х	Х	Х	Х
b	Х			Х	Х	Х	Х
с			Х	Х	Х	Х	Х
d		Х		Х	Х	Х	Х
e			Х	Х			
f			Х		Х		
g			Х			Х	
h			Х			Х	
i	Х	Х		Х			
j	Х	Х			Х		
k	Х	Х				Х	
l	Х	Х					Х
m	Х		Х				
n				Х	Х		
0						Х	X
р		Х	Х				

Tableau 5.1 – FDI à l'aide des relations de redondance analytique

Les résidus des relations de redondance sont obtenues en soustrayant les deux côtés des égalités des équations 5.3. Le seuillage de ces résidus permet la détection et l'isolement de défauts des éléments (capteur et commande) contribuant à l'élaboration de ces équations. Le tableau 5.1 représente une table de vérité pour la FDI des défauts à l'aide du seuillage des relations de parité. Les résidus sont numérotés de 'a' à 'p' comme dans les équations 5.3, le croix x signifie un dépassement.

Un défaut sur chacun des éléments de la première ligne induit le dépassement du seuil d'une combinaison de résidus. L'identification de la combinaison de résidus dépassant leurs seuils aboutira à la FDI de défauts.

Une phase supplémentaire de filtrage par un filtre passe bas (Butterworth) permet de réduire les oscillations indésirables dans les résidus.

Le défaut injecté sur les mesures des capteurs de localisation (GPS, IMU, odomètres) prend la forme d'un bruit gaussien additif de moyenne  $\mu = 0$  et d'un écart type  $\sigma = 0.5$ . Sa fonction de densité de probabilité est écrite dans l'équation 5.5.

$$[H]P(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$
(5.5)

La mesure du laser, ou le scan, est fourni sous la forme d'un vecteur contenant la distance parcourue par chaque rayon lumineux avant de se refléter. Pour générer le défaut laser, le même type de défaut est utilisé. Le bruit est ajouté à ce vecteur impliquant une perturbation sur la mesure. Le défaut de perte de paquets de communication UWB est simulé dans le programme de réception de paquets. Lorsqu'un défaut est actif, le programme ignore le paquet et ne le prend pas en considération.

En ce qui concerne le blocage de roues, le modèle cinématique du robot prenant en compte l'effet de ce type de défaut est présenté dans l'équation 5.13. Une erreur sur chaque roue entraînera des valeurs différentes de  $\beta_v$ ,  $\beta_w$ ,  $\beta_{v2}$  et  $\beta_{w2}$ .



FIGURE 5.7 – Forces et moments

En considérant la dynamique du robot à la figure 5.7,  $\overrightarrow{F_i}$  et  $\overrightarrow{R_i}$ , sont des forces produites respectivement actives et réactives (résultant du frottement). En appliquant la loi de Newton, lorsqu'une des roues n'est pas bloquée :

$$\sum (\overrightarrow{F_i} + \overrightarrow{R_i}) = \overrightarrow{F} + \overrightarrow{R} = m.\overrightarrow{a_n}$$
(5.6)

où  $\overrightarrow{a_n}$  correspond à l'accélération normale du robot (sans défaut).

Si la roue *i* est bloquée,  $F_i$  sera mis à zéro et  $R_i$  augmentera de  $r_0$ , à une valeur inconnue. Une équation modifiée projetée sur  $\vec{X}$  conduira à :

$$\alpha_1 F + \alpha_2 R = m.a_b \tag{5.7}$$

Où  $\alpha_1$  et  $\alpha_2$  sont des paramètres inconnus et  $a_b$  est la valeur d'accélération du robot pour les mêmes entrées, sous blocage de roue. En soustrayant l'équation (5.7) et l'équation (5.6), on obtient :

$$F(\alpha_1 - 1) + R(\alpha_2 - 1) = m(a_b - a_n)$$
(5.8)

En organisant l'équation 5.8, on obtient après intégration :

$$v_b = \frac{\int (F(\alpha_1 - 1) + R(\alpha_2 - 1))dt}{m} + v_n$$
(5.9)

Où  $v_n$  et  $v_b$  sont la vitesse linéaire du robot en mode normal et la valeur correspondante en mode roue bloquée. Sans perte de généralité, en supposant un signal de commande constant, les forces actives et réactives deviennent des constantes. F et R dépendent des signaux de commande v et w. Ainsi, on peut écrire :

$$v_b = \beta_v v + \beta_w w + v_n \tag{5.10}$$

De même, à partir de l'équation :

$$2L(\overrightarrow{F_r} - \overrightarrow{F_l} + \overrightarrow{R_l} - \overrightarrow{R_r}) = \overrightarrow{M} = I \overrightarrow{\theta}$$
(5.11)  
Avec :  

$$\overrightarrow{F_r} = \overrightarrow{F_1} + \overrightarrow{F_4}; \ \overrightarrow{F_l} = \overrightarrow{F_2} + \overrightarrow{F_3}$$

$$\overrightarrow{R_r} = \overrightarrow{R_1} + \overrightarrow{R_4}; \ \overrightarrow{R_l} = \overrightarrow{R_2} + \overrightarrow{R_3}$$

Où  $\overrightarrow{M}$  le couple résultant, I la matrice inertielle du robot et  $\overrightarrow{\theta}$  l'accélération angulaire. En suivant le même raisonnement, on obtient :

$$w_b = w_n + \beta_{w2}w + \beta_{v2}v \tag{5.12}$$

Ainsi, en implémentant les équations des paramètres de défaut 5.10 et 5.12 dans le modèle cinématique du robot (équation 5.1) et en considérant  $v_n = v$  et  $w_n = w$ , le modèle du robot correspondant au défaut de blocage d'une roue est donné par les équations 5.13.

$$x_{t+1} = x_t + \cos(\theta_t)(v + v\beta_v + w\beta_w)dt$$
  

$$y_{t+1} = y_t + \sin(\theta_t)(v + v\beta_v + w\beta_w)dt$$
  

$$\theta_{t+1} = \theta_t + (w(1 + \beta_{w2}) + v.\beta_{v2})dt$$
(5.13)

Ce modèle est adopté pour les défauts de blocage de roue. Les paramètres de défaut à estimer sont  $\beta_v$ ,  $\beta_{v2}$ ,  $\beta_w$  et  $\beta_{w2}$ .

Cette manière d'injecter un défaut est valable en simulation et sur le robot réel. Toutefois, la simulation sera privilégiée afin d'éviter l'échauffement des moteurs; conséquence directe d'un blocage de roue.

## 5.3 Mise en œuvre de S2D2R

Comme décrit à la figure 3.2, pour définir la MHD, les méthodes sont évaluées et comparées entre elles pour associer à chaque défaut, la ou les méthodes les plus performantes.

La prochaine section montre les résultats d'application des méthodes individuelles de diagnostic avant de les comparer et de choisir les plus performantes en se servant de l'indicateur de performance défini dans 3.2.2.

### 5.3.1 Définition de la version initiale de la MHD

#### 5.3.1.1 Défauts des capteurs de localisation

**Application des Filtres de Kalman** Les deux versions étendues (EKF) et unscented (UKF) du filtre de Kalman sont appliquées tel que décrit dans 4.1.3. Ces filtres implémentent le modèle Newtonien du robot. Les paramètres des filtres sont comme suit :

- La position initiale  $(x_0, y_0, \theta_0)$  est déterministe. Elle correspond à la position réelle du robot. La matrice de covariance initiale de l'estimation  $P_0$  est choisie empiriquement d'après plusieurs tests :  $P_0 = (0.01 \ 0.01 \ 0 \ 0.01 \ 0.01 \ 0 \ 0 \ 0.01)$ .
- La matrice de covariance de mesure est choisie conformément aux bruits des capteurs utilisés en simulation. Elle est diagonale ; ce qui correspond à des variables d'états indépendantes. La matrice est :  $R = (0.01 \ 0 \ 0 \ 0.01 \ 0 \ 0 \ 0.01)$ .

— La matrice de covariance du processus est calculée numériquement à l'aide de l'algorithme génétique présenté dans le chapitre précédent. À chaque lancement de l'algorithme, nous obtenons des valeurs différentes. Après avoir réalisés différents tests, nous avons choisi :

Pour l'EKF :  $Q_{EKF} = (0.03447 \ 0 \ 0 \ 0.12766 \ 0 \ 0 \ 0.25896),$ 

Pour l'UKF :  $Q_{UKF} = (0.06421 \ 0 \ 0 \ 0.07257 \ 0 \ 0 \ 0.07956).$ 

La figure 5.8 représente la distance de Mahalanobis (résidu, en vert) générée par l'EKF sur le pose-provider PP2 (fusion des données IMU, odomètre et GPS). La ligne horizontale interrompue (rouge) est le seuil de détection (ici à 0.2). L'axe des abscisses représente le temps. Dans cette simulation, nous introduisons un défaut (bruit avec un écart type de 0.1) sur une période de 0,5s toutes les 1s. Les lignes pointillées verticales définissent le début et la fin de l'injection du défaut (ici quatre injections sont représentées). Le seuil de détection est fixé à 0.2. On remarque clairement que cet EKF détecte les perturbations répétitives et intermittentes introduites. L'oscillation du résidu lors de la détection du défaut est une conséquence de la nature aléatoire du défaut.

La figure 5.9 illustre l'isolement de défauts GPS à l'aide de cette méthode. Les points en vert (reliés par la courbe verte) représentent la réponse de la méthode. La ligne interrompue en rouge représente l'état réel, défectueux ou normal. Les deux lignes interrompues en vert et en bleu représentent les signaux de commandes linéaire et angulaire. Cette figure montre que la méthode isole de manière précise et fiable les défauts GPS.

La même méthode a ensuite été testée sur d'autres défauts (deux filtres, un sur chaque PP, restant sur des données de haut niveau). Le glissement, le blocage d'une roue et la roue libre (moteur passif) sont détectables seulement en fournissant au robot une commande de virage relativement rapide (v > 1.5m/s; w > 1rd/s). Le délai de détection du glissement peut être supérieur à 3s selon la commande. Le tableau 5.2 résume les résultats obtenus par le test de cette méthode sur ces trois défauts.



FIGURE 5.8 – Détection des perturbations IMU par EKF sur PP2



FIGURE 5.9 – Détection de défauts GPS par la méthode de 2 EKF

	UKF sur PP1	UKF sur PP2	EKF sur PP1	EKF sur PP2
Glissement	Pas de détection	Détection lors	Pas de détection	Détection lors
		du virage, délai		du virage, délai
		de détection		de détection
		> 3s		> 3s
Roue bloquée	Uniquement à	Uniquement à	Uniquement à	Uniquement à
	grande vitesse	grande vitesse	grande vitesse	grande vitesse
Roue libre	Uniquement à	Uniquement à	Uniquement à	Uniquement à
	grande vitesse	grande vitesse	grande vitesse	grande vitesse

Tableau 5.2 – Détection des défauts de glissement, roue bloquée et roue libre à partir des filtres EKF et UKF

**Application des relations de redondance** Les figures 5.10 montrent la FDI de défauts GPS et IMU à l'aide de la redondance analytique. Dans 5.10a, la méthode isole les défauts GPS mais met trop de temps pour se rétablir après la disparition du défaut. Dans le cas de perturbations IMU, l'algorithme ne réussit pas à isoler les défauts comme montré à la figure 5.10b. Nous remarquons des détections dont la nature du défaut est inconnue (*Unknown* sur la figure), ce qui correspond à une combinaison de dépassements de résidus non connue.

Théoriquement, la surveillance des relations de redondance analytique (ou de parité) permet la FDI des défauts de perturbations IMU et GPS. Mais en appliquant cette méthode, les résultats n'étaient pas comparables à ceux des autres méthodes indiquées dans cette section.

**Application de l'apprentissage profond** Les deux modèles expliqués dans 4.6.1 sont appliqués sur les résidus. Ces deux modèles C et E-C ont bien réagi à ces deux défauts comme montré à la figure 5.11.



(b) IMU

FIGURE 5.10 – FDI à l'aide du seuillage des équations de la redondance analytique



 ${\rm Figure}~5.11$  – Résultats de l'application des modèles de l'apprentissage profond sur les défauts GPS et IMU

Le filtre de Butterworth utilisé pour filtrer les données d'apprentissage est plus sévère que celui utilisé en temps réel pour la classification. Le premier a une fréquence de coupure de 2Hz et il est du 3ème degré tandis que le deuxième a une fréquence de coupure de 3Hz et il est du 2ème degré. Ceci favorise davantage la détection rapide des défauts en augmentant la sensibilité des algorithmes aux perturbations.

Le traitement par seuillage des résidus de la redondance analytique ne permet pas d'avoir une performance acceptable en FDI. En fait, le seuillage simple de ces résidus ne permet pas d'en tirer toutes les informations utiles contrairement aux modèles de l'apprentissage profond. Un défaut n'apparaîtrait pas forcement sur les résidus comme une pure augmentation de valeurs, il peut apparaître comme une succession de changement sur un ou plusieurs résidus.

### 5.3.1.2 Défauts laser

**Application du filtres de Kalman** Pour la détection de défauts de perturbations du laser, la méthode expliquée dans 4.1.4 est mise en œuvre. Le laser 2d utilisé est de type Sick. Il possède les caractéristiques suivantes :

- Distance minimale de l'objet détectable : 5cm
- Distance maximale de l'objet détectable : 10m
- Champ de détection :  $[-135^\circ, +135^\circ]$
- Incrémentation d'angle (ou résolution) : 1 degré



FIGURE 5.12 – Le robot entouré par 4 objets connus

La scène est composé de 4 objets connus, placés aux points  $(\pm 2, \pm 2)$  comme montré à la figure 5.12. L'EKF utilisé pour la détection de perturbations possède 12 éléments dans son vecteur d'état dont 3 pour chaque objet connu. Leurs caractéristiques sont comme suit :

- La position initiale  $(x_0, y_0, \theta_0)$  est déterministe. Elle correspond à la position réelle des objets connus par rapport au robot. La matrice de covariance initiale de l'estimation  $P_0$  est choisie empiriquement d'après plusieurs tests. C'est une matrice de taille  $12 \times 12$ . Ses éléments sont tous nuls sauf les éléments sur le diagonale qui valent 0.1
- La matrice de covariance de mesure est choisie conformément aux bruits des capteurs utilisés en simulation. C'est une matrice de taille  $12 \times 12$ . Ses éléments sont tous nuls sauf

les élément sur le diagonale qui valent 0.01.

 La matrice de covariance du processus est calculée numériquement à l'aide de l'algorithme génétique. Elle est de taille 12 × 12.

	(1.77	0.66	0.01	1.54	0.95	0.02	1.74	1.48	1.46	0.	0.04	1.47	
	0.35	1.2	1.84	0.73	1.99	1.81	1.37	0.50	0.08	0.49	0.799	0	
	0.	1.79	0.88	0.87	0.	1.38	0.84	1.16	0.82	1.30	1.43	0.97	
	0.	0.52	0.93	1.76	1.15	1.83	0.004	0.10	1.94	1.94	1.58	0.73	
	1.82	1.18	0.	1.01	1.02	1.54	1.76	0.47	1.28	1.47	0.023	0.54	
0 -	1.79	0.13	0.34	0.09	0.64	0.13	1.48	0.96	0.15	0.11	1.67	0.58	
Q =	1.05	1.34	0.07	1.57	1.8	1.28	1.04	0.33	1.01	1.99	1.55	1.42	
	1.70	0.02	1.15	0.71	0.16	0.20	1.47	0.60	1.46	1.04	1.10	0.51	
	1.14	0.09	1.22	0.78	0.86	0.25	0.03	1.30	0.82	1.81	0.41	1.62	
	0.93	1.74	1.61	0.02	1.56	0.91	0.97	1.90	1.81	0.01	1.35	0.64	
	1.97	1.47	0.08	1.93	0.01	0.16	0.53	1.89	1.56	1.95	1.26	0.18	
	0.37	1.54	1.54	0.10	0.02	0.49	0.54	1.71	1.78	0.94	1.49	1.36/	
												(5	.14)

L'absence d'un objet de la scène peut être rapidement isolé. Quant à la détection de perturbation du signal, la figure 5.13 montre la réponse de l'EKF aux perturbations. Les croix vertes représentent la réponse de la méthode. Les lignes interrompues en rouge représentent l'état réel, défectueux ou normal. Les deux lignes interrompues en vert et en bleu représentent les signaux de commandes linéaire et angulaire. Les perturbations sont détectées efficacement.



FIGURE 5.13 – Détection de la perturbation du laser par l'EKF

#### 5.3.1.3 Défauts de communication par UWB

La méthode expliquée dans le paragraphe 4.7 est appliquée pour le cas du suivi de robot. Trois tags (nœuds) sont placés sur chacun des robots suiveur et leader. Ils servent à transmettre les informations nécessaires au suivi du robot. i.e. la distance entre les deux robots (calculée selon le temps de transmission et de réception d'un message), la position dans le repère absolu du robot leader et la commande en vitesse du robot leader. À chaque instant t, un seul tag initiateur I

'discute' avec un seul tag récepteur R. Un séquenceur s'occupe de l'organisation de l'ordre de communication.



FIGURE 5.14 – Contexte d'application de la communication UWB

Comme illustré à la figure 5.14, chaque tag initiateur communique itérativement avec chacun des tags récepteurs. La matrice instantanée d'état de communication M est de la forme :

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

Les deux vecteurs de confiance ont la forme :  $T = \begin{pmatrix} t_1 & t_2 & t_3 \end{pmatrix}$ .

Une succession de modes est définie pour tester l'algorithme dans des conditions différentes. Elle est composée de 11 modes dont le premier et le dernier sont le fonctionnement normal. On bascule entre mode normal et défaut à chaque fois. la durée d'une mode est d'environ 30s. Les modes sont :

- 1. Normal.
- 2. Communication impossible entre  $R_0$  et  $I_1$ .
- 3. Normal.
- 4.  $R_0$  ne fonctionne pas.
- 5. Normal.
- 6. Perte de données sur  $R_0$  de faible fréquence.
- 7. Normal.
- 8.  $I_2$  ne fonctionne pas.
- 9. Normal.
- 10.  $I_2$  et  $R_0$  ne fonctionnent pas.
- 11. Normal.

L'algorithme a réussi à isoler la totalité de ces défauts. Il est même capable de détecter deux défauts simultanés comme le cas 10.

#### 5.3.1.4 Défauts de blocage de roues

Le défaut de blocage de roues est traité par quatre algorithmes. Les résultats de ces algorithmes sont montrés dans les paragraphes qui suivent.

mode	$\beta_v$	$\beta_w$	$\beta_{v2}$	$\beta_{w2}$
n	0.03	-0.08	-0.013	-0.3
Blocage roue avant droite	0.05	-0.3	-0.55	-0.2
Blocage roue avant gauche	0	-0.3	0.5	-0.2
Blocage roue arrière gauche	0	-0.2	0.7	-0.6
Blocage roue arrière droite	-0.2	-0.3	-0.85	-0.7

Tableau 5.3 – Les paramètres identifiés

En ce qui concerne la modélisation de ce défaut, les paramètres de l'équation 5.13 sont estimés hors ligne grâce à une base de données construite en simulation. Les valeurs correspondant à chaque mode sont indiqués dans le tableau 5.3. En raison de la configuration mécanique du robot, il est clair dans la table 5.3 que le blocage des roues du même côté (droite ou gauche) ont un effet similaire sur les paramètres  $\beta_{v2}$  et  $\beta_{w2}$ , tandis que le blocage de roue arrière est plus puissant. Cela signifie que le blocage d'une roue arrière a un effet plus important sur la dynamique du robot que le blocage d'une roue avant.

**Application des relations de redondance analytique** Contrairement aux méthodes basées sur l'estimation de paramètres, l'application de cet algorithme ne nécessite pas une connaissance des paramètres des défauts.



FIGURE 5.15 – Résultats de l'application de la redondance analytique sur le blocage des roues

Comme l'indique le tableau 5.1, la surveillance des résidus issus de la redondance analytique permet la FDI des défauts des roues. En effet, un blocage sur l'une des roues modifie les résidus auxquels contribue la valeur de sa vitesse angulaire. Après un filtrage utilisé pour réduire l'effet du bruit, le seuillage a permis de détecter les défauts de blocage des roues, mais la performance d'isolement n'est pas optimale.

La figure 5.15 illustre les réponses de l'algorithme face aux quatre défauts successifs de blocage des roues.

Cet algorithme isole les défauts mais subit des fausses alarmes pour les défauts de blocage des roues avant. Tandis que pour les roues arrières, les défauts sont détectés mais ne sont pas bien isolés. De plus, on note la présence des fausses alarmes.

En conclusion, le seuillage des relations de redondance n'est pas efficace pour isoler ces défauts. Ces résidus doivent être traités d'une manière plus complexe afin d'en extraire les informations utiles.

Application de l'estimation récursive de paramètres Suite à l'estimation hors lignes des paramètres  $\beta$  de l'équation 5.13, la même base de données est réutilisée avec un RLS hors ligne pour créer une autre base contenant les paramètres estimés récursivement pour chaque mode. Cette dernière sert à entraîner l'algorithme de LDA comme à la figure 4.5. Les résultats de l'application de cette méthode pour la FDI des défauts de blocage des roues sont montrés à la figure 5.16.



FIGURE 5.16 – Résultats de l'application de l'estimation récursive de paramètres sur le blocage des roues

D'après ces figures, la performance de détection et d'isolement des roues arrières est meilleure

que celle des roues avant c.à.d. qu'il apparaît moins de détections ratées pour les défauts sur les roues arrières. Les deux cas ne présentent pas de fausses alarmes.

En conclusion, cet algorithme est robuste aux fausses alarmes, mais il met du temps avant de détecter ou isoler le défaut.

**Application de la MMAE** Le schéma de MMAE expliqué à la section 4.4 est appliqué pour l'isolement de défauts de blocage des roues. Le paramétrage de l'UKF est fait empiriquement pour  $P_0$ , à l'aide de l'algorithme génétique proposé au paragraphe 4.1.2 pour Q et selon les caractéristiques des capteurs pour R. Ces matrices sont définies comme suit :

$$-P_0 = \begin{pmatrix} 0.07 & 0 & 0 \\ 0 & 0.08 & 0 \\ 0 & 0 & 0.09 \end{pmatrix}$$
$$-Q = \begin{pmatrix} 0.064 & 0 & 0 \\ 0 & 0.08 & 0 \\ 0 & 0 & 0.09 \end{pmatrix}$$
$$-R = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

La taille de la fenêtre de blocage des résidus avant de les filtrer et de les traiter correspond à 30 itérations.



FIGURE 5.17 – Résultats de l'application de MMAE sur le blocage des roues

La figure 5.17 montre la réponse de cet algorithme face aux défauts de blocage des roues. Les défauts de blocage des roues arrières sont mieux isolés que ceux des roues avants. Certaines fausses alarmes apparaissent également.

**Application du filtre à particules** Le résultat de l'application du PFM fusionnant les deux algorithmes PF et MMAE montre que a détection de défauts est plus rapide que celle de RPE tout en étant plus robuste aux fausses alarmes que MMAE. Le rétablissement de la méthode au mode normal après la fin d'un défaut semble rapide.



FIGURE 5.18 – Résultats de l'application de PFM sur le blocage des roues

La figure 5.18 montre la réponse du PFM face aux blocages des roues. MMAE et RPE ont le même paramétrage que les algorithmes des paragraphes précédents. Le nombre de particules est fixé à 40. Comme pour les deux algorithmes précédents, l'isolement des défauts sur les roues arrières est plus performante que celle sur les roues avants.

**Application de l'apprentissage profond** Après l'entraînement, les deux modèles de l'apprentissage profond sont appliqués sur les défauts de blocage des roues.

La figure 5.19 montre la réponse des deux modèles proposés face aux défauts de blocage des roues. Les quatre premières figures présentent les résultats du Classificateur (C). Les autres représentent ceux de l'encodeur suivi d'un classificateur (EC).

Ces résultats montrent que les deux algorithmes ont su diagnostiquer ces défauts. Le Classificateur présente moins de fausses alarmes par rapport à EC, tandis que EC semble détecter et isoler les défauts plus rapidement.

Pour évaluer puis comparer objectivement ces résultats, nous allons appliquer l'indicateur de performance proposé dans 3.2.2. Cette évaluation, nous permettra de définir une forme initiale de la distributions des méthodes dans la MHD.



(a) C sur blocage de la roue avant droite



(c) C sur blocage de la roue arrière droite



(e) E-C sur blocage de la roue avant droite



(g) E-C sur blocage de la roue arrière droite



(b) C sur blocage de la roue avant gauche



(d) C sur blocage de la roue arrière gauche



(f) E-C sur blocage de la roue avant gauche



(h) E-C sur blocage de la roue arrière gauche

 ${\rm FIGURE}$  5.19 – Résultats de l'application des modèles de l'apprentissage profond sur le blocage des roues

## 5.3.1.5 Évaluation de la performance des méthodes de diagnostic

L'indicateur de performance proposé dans la section 3.2.2 est appliqué sur l'ensemble des méthodes de diagnostic testées. Il est calculé deux fois pour chaque méthode. À chaque fois, un coefficient (ou poids) différent est attribué aux éléments de l'indicateur lors de la somme pondérée conformément au tableau 3.1 :

- 1. En favorisant la détection de défauts ou la sensibilité : Dans ce cas, les éléments Taux de Détection Ratée (MDR) et Délai de Détection (DD) ont un coefficient (ou poids) de 0.4 chacun contre 0.2 pour le Taux de Fausse Alarme (FAR) dans le sous indicateur de détection. Quant au sous indicateur d'isolement, le Taux de Faux Isolement (FIR) et le Délai d'Isolement (ID) ont un coefficient de 0.4 contre 0.2 pour le Délai de rétablissement du mode normal (NMRD)
- 2. En favorisant la robustesse contre les fausses alarmes : Pour calculer le sous indicateur de détection, un coefficient de 0.6 est attribué à FAR contre 0.2 pour le MDR et le DD. Concernant le sous indicateur d'isolement, un coefficient de 0.6 est attribué à NMRD contre 0.2 pour le FIR et l'ID.

Dans la suite de ce paragraphe, l'indicateur de performance est calculé pour toutes les méthodes diagnostiquant chaque défaut. Les sous indicateurs de détection et d'isolement d'une méthode dépendent du défaut considéré, tandis que les deux autres sous indicateurs, à savoir ceux du "coût" et "prévision", ne dépendent pas du défaut. L'évaluation des sous indicateurs de coût et prévision pour les méthodes de diagnostic considérées est décrite dans le tableau 5.4.

		Coût			Prévi	ision
Méthode	TPC	RDS	nb défauts	Non prévus	EE	Adaptabilité/10
EKF	$10^{-3}$	0	3	1	1	9
Relations	$10^{-4}$	0	7	1	0	7
С	$10^{-3}$	20,000pts/mode	7	0	1	1
EC	$10^{-3}$	20,000pts/mode	7	0	1	1
MMAE	$5.10^{-3}$	100 points/mode	4	1	1	3
RPE	$10^{-4}$	300 points/mode	4	0	0	3
PF	$10^{-2}$	300 points/mode	4	1	1	2
Vecteur de confiance	$10^{-3}$	0	6	0	1	10
EKF-laser	$10^{-3}$	0	2	1	1	9

Tableau 5.4 – Valeurs des sous indicateurs indépendants des défauts

Le temps de calcul par cycle présenté dans ce tableau correspond au temps moyen d'exécution par cycle calculé à la fin du scénario. Le temps de calcul de toutes les méthodes abordées permet leur exécution avec une fréquence de 50 Hz. Le PF est le plus gourmand puisqu'il fusionne deux autres algorithmes. Les relations de redondance sont les moins gourmandes, juste devant les filtres de Kalman. Les modèles de l'intelligence artificielle ont un temps de calcul comparable à celui des filtres de Kalman.

En ce qui concerne la quantité de données à stocker, la méthode des deux KF, les relations de la redondance analytique, le vecteur de confiance et l'EKF sur le laser ne requièrent pas de stocker des données. Au contraire, les autres méthodes proposées exigent de stocker des données pour l'entraînement. Les méthodes qui en demandent le plus sont les deux modèles de

l'apprentissage profond où 20000 lignes d'entrée par mode considéré sont nécessaires pour la phase d'apprentissage. Les méthodes basées sur l'estimation de paramètres ont besoin d'environ 100 points par mode considéré afin d'estimer hors ligne les paramètres des défauts. Enfin, RPE et PF ont besoin de 200 points supplémentaires par mode afin d'entraîner le LDA.

Le nombre de défauts (considérés) traités pour chaque méthode est indiqué dans ce tableau (tableau 5.4). Les deux modèles de l'IA (C et EC) et les relations de redondance profitent du plus grand nombre de défauts isolables. Une attention particulière doit être portée aux deux méthodes EKF sur laser et vecteur de confiance. L'EKF sur laser isole les défauts qui pourraient se produire dans un scan laser, à savoir, des perturbations du signal et l'absence d'un objet connu. Quant au vecteur de confiance, les défauts isolés dans ce cas sont les problèmes de communication entre les tags. Le type de défauts est unique mais les sources en sont multiples (6).

Les méthodes basées sur le filtrage de Kalman peuvent détecter les perturbations causées par des défauts imprévus sans savoir leur source. C'est le cas aussi de la redondance analytique et PFM qui englobe des filtres de Kalman. Par contre, le comportement du RPE et de l'apprentissage profond est imprévisible au cas de défaut imprévu dès le départ.

Certaines méthodes fournissent une estimation de la fiabilité de la décision ou la probabilité que cette décision soit erronée. Dans les cas des méthodes dont la décision est prise grâce au seuillage d'un signal, la distance entre ce signal et le seuil est une indication de fiabilité de la décision. Plus la distance est grande plus la décision est sûre. Les algorithmes de l'intelligence profond fournissent aussi une estimation sur la probabilité de chaque mode. Quant au PF, le nombre de particules appartenant à chaque mode construit une estimation sur la probabilité d'erreur.

L'adaptation de la méthode de vecteur de confiance est très simple car la conception est facile. Il suffit d'adapter les vecteur et la matrice d'état.

Pour appliquer les méthodes basées sur le KF, il suffit d'adapter le modèle et de reparamétrer le filtre. Généralement pour des applications similaires, le paramétrage ne change pas beaucoup.

Pour appliquer le MMAE et le RPE, il va falloir refaire la modélisation du défaut et recueillir les données nécessaires à l'estimation des paramètres. L'adaptation du PF est encore plus complexe, puisqu'un paramétrage supplémentaire des différents gains est nécessaire.

Quant à la redondance analytique, les relations entre les différents éléments du robot doivent être déduites avant d'en tirer les résidus utiles.

Finalement, l'adaptation des méthodes de l'apprentissage profond est la plus difficile. Pour appliquer ces méthodes sur un autre cas, il faudra d'abord redéfinir le réseau selon le besoin, puis construire la base de données nécessaire pour l'entraînement. L'entraînement est une opération longue qui, selon le nombre de modes et la quantité de données, peut prendre des jours pour s'accomplir.

	Déte	ction	Isole	ment
Méthode	Dét fav	Rob fav	Dét fav	Rob fav
EKF	0.92	0.95	0.93	0.96
Relations	0.85	0.73	0.73	0.42
С	0.88	0.93	0.86	0.87
EC	0.9	0.9	0.86	0.87

Tableau 5.5 – Évaluation des méthodes de diagnostic, cas du défaut GPS

**Diagnostic des défauts GPS** Dans tous les cas, pour la détection de défauts GPS, la méthode des deux filtres de Kalman s'avère la plus performante juste devant le EC.

Le tableau 5.5 résume les résultats de l'évaluation des méthodes de diagnostic traitant le défaut de GPS.

Notons que les deux versions du filtre de Kalman EKF et UKF ont des résultats très proches voire quasi-identiques dans ce cas. Nous avons choisi de privilégier l'EKF puisqu'il est moins gourmand que l'UKF en terme de temps de calcul.

**Diagnostic des défauts IMU** L'EKF s'avère l'algorithme le plus performant pour détecter un défaut IMU. Les deux modèles de l'apprentissage profond, C et EC, sont les plus performants pour l'isoler. L'EKF détecte donc qu'un défaut est survenu plus rapidement que C et EC, mais ces derniers sont plus précis en terme d'isolement. Le tableau 5.6 résume les résultats de l'évaluation des méthodes de diagnostic traitant un défaut de l'IMU.

	Déte	ection	Isole	ment
Méthode	Dét fav	Rob fav	Dét fav	Rob fav
EKF	0.959	0.97	0.89	0.85
Relations	0.24	0.46	0	0
С	0.94	0.96	0.93	0.9
EC	0.93	0.92	0.93	0.9

Tableau 5.6 – Évaluation des méthodes de diagnostic, cas du défaut IMU

Pour ce défaut aussi, EKF et UKF ont des résultats quasi-identiques. Nous avons adopté l'EKF puisque son temps de calcul est réduit en le comparant à l'UKF.

**Diagnostic du blocage de la roue avant gauche** Pour le défaut de blocage de la roue avant gauche, le Classificateur (C) est la meilleure méthode en terme de détection. Pour l'isolement, EC est le plus performant. Le tableau 5.7 résume les résultats de l'évaluation des méthodes de diagnostic traitant le défaut de la roue avant gauche.

	Déte	ction	Isole	ment
Méthode	Dét fav	Rob fav	Dét fav	Rob fav
EKF	0.2	0.6	0	0
Relations	0.85	0.79	0.86	0.82
MMAE	0.48	0.73	0.139	0.07
RPE	0.32	0.65	0.25	0.54
PF	0.44	0.7	0.23	0.36
С	0.94	0.96	0.9	0.9
EC	0.92	0.9	0.93	0.9

Tableau 5.7 – Évaluation des méthodes de diagnostic, cas du défaut Roue avant gauche

**Diagnostic du blocage de la roue avant droite** Pour le défaut de blocage de la roue avant droite, le Classificateur (C) est la meilleure méthode en terme de détection. Pour l'isolement, C et EC ont la même performance. Le tableau 5.8 résume les résultats de l'évaluation des méthodes de diagnostic traitant le défaut de la roue avant droite.

	Détection		Isole	ment
Méthode	Dét fav	Rob fav	Dét fav	Rob fav
EKF	0.2	0.6	0	0
Relations	0.94	0.84	0.98	0.98
MMAE	0.5	0.74	0.25	0.37
RPE	0.344	0.66	0.2	0.32
PF	0.43	0.7	0.25	0.43
C	0.95	0.97	0.93	0.9
EC	0.93	0.91	0.93	0.9

Tableau 5.8 – Évaluation des méthodes de diagnostic, cas du défaut Roue avant droite

**Diagnostic du blocage de la roue arrière gauche** Pour le défaut de blocage de la roue arrière gauche, le Classificateur (C) est la meilleure méthode en terme de détection. Pour l'isolement, C et EC sont équivalents dans le cas où la détection est favorisée durant le calcul de l'indicateur. Dans le cas où la robustesse est favorisée, EC sera plus performant que C. Le tableau 5.9 résume les résultats de l'évaluation des méthodes de diagnostic traitant le défaut de la roue arrière gauche.

	Détection		Isole	ment
Méthode	Dét fav	Rob fav	Dét fav	Rob fav
EKF	0.2	0.6	0	0
Relations	0.94	0.84	0.87	0.92
MMAE	0.77	0.86	0.35	0.35
RPE	0.54	0.72	0.43	0.25
PF	0.63	0.78	0.35	0.34
С	0.95	0.96	0.93	0.9
EC	0.93	0.9	0.93	0.91

Tableau 5.9 – Évaluation des méthodes de diagnostic, cas du défaut Roue arrière gauche

**Diagnostic du blocage de la roue arrière droite** Pour le défaut de blocage de la roue arrière droite, le Classificateur (C) est la meilleure méthode que ce soit pour la détection ou bien pour l'isolement. Le tableau 5.10 résume les résultats de l'évaluation des méthodes de diagnostic traitant le défaut de la roue arrière droite.

	Déte	ection	Isole	ment
Méthode	Dét fav	Rob fav	Dét fav	Rob fav
EKF	0.2	0.6	0	0
Relations	0.94	0.84	0.2	0.59
MMAE	0.63	0.79	0.33	0.34
RPE	0.53	0.71	0.32	0.26
PF	0.58	0.75	0.41	0.4
С	0.95	0.96	0.93	0.9
EC	0.93	0.9	0.83	0.86

Tableau 5.10 – Évaluation des méthodes de diagnostic, cas du défaut Roue arrière droite

**Diagnostic des défauts laser** La performance de la méthode de détection des défauts laser à l'aide d'un EKF est montrée sur le tableau 5.11 (cas de quatre objets connus). D'après ce tableau, les résultats sont admissibles.

	Déte	ction	Isole	ment
Méthode	Dét fav Rob fav		Dét fav	Rob fav
EKF	0.85	0.92	0.85	0.92

Tableau 5.11 – Évaluation des méthodes de diagnostic, cas de défauts laser

Cette méthode permet de diagnostiquer ce type de défaut sans problèmes.

Les valeurs de l'indicateur de performance sont identiques en détection et en isolement puisque cette méthode est dédié uniquement au défaut de perturbations laser.

**Diagnostic des défauts UWB** La performance de la méthode de détection des défauts de communication des tags UWB à l'aide de la méthode de vecteurs de confiance est montrée dans le tableau 5.12. Les résultats sont admissibles. Cette méthode peut être alors adoptée pour surveiller la communication. Bien que cette méthode diagnostique uniquement les défauts UWB, les valeurs des indicateurs de performance en détection et en isolement ne sont pas identiques puisque plusieurs défauts possibles sont considérés (entre  $i_n$  et  $r_j$ ).

	Déte	ection	Isolement		
Méthode	Dét fav	Rob fav	Dét fav	Rob fav	
Vecteur de confiance	0.85	0.89	0.73	0.67	

Tableau 5.12 – Évaluation des méthodes de diagnostic, cas de défauts UWB

#### 5.3.1.6 Association des défauts aux méthodes retenues

À la lumière des résultats obtenus dans le paragraphe précédent (5.3.1.5), nous définissons dans ce paragraphe une forme initiale des méthodes de diagnostic construisant la MHD. Les résultats de l'indicateur favorisant la détection sont utilisés.

D'abord, pour détecter et isoler les défauts GPS, la méthodes des deux filtres de Kalman a les plus grandes valeurs de performance.

Dans le cas de l'IMU, la méthode des deux filtres est la meilleure en détection mais elle ne l'est pas en isolement. Ce sont les deux modèles de l'apprentissage profond qui donnent les meilleurs résultats. KF est adopté pour la détection mais une fois le défaut détecté, c'est l'apprentissage profond qui sera utilisé pour isoler un défaut IMU.

Pour le blocage des roues, le modèle C de l'apprentissage profond a les meilleures performances en détection et en isolement. Cette méthode sera choisie pour diagnostiquer les défauts de blocage des roues.

Les défauts UWB et laser sont diagnostiqués respectivement par les méthodes de vecteurs de confiance et l'EKF. Ces deux méthodes ont une performance acceptable d'après leur évaluation. La figure 5.20 décrit la distribution initiale des méthodes de diagnostic à mettre en œuvre dans la MHD.



FIGURE 5.20 – Distribution initiale des méthodes de diagnostic de la MHD (MHD<sub>0</sub>)

En ce qui concerne les autres types de défaut, comme le glissement, ils seront détectés mais non isolés. Il appartient à l'opérateur dans un premier temps de les isoler en temps réel à l'aide des informations fournies par la MHD. Puis, comme montré à la figure 3.4, ses décisions mettront à jour la base de connaissances, ce qui permettra l'isolement de ces défauts automatiquement dans le futur.

Rappelons que ce schéma risque de changer à tout moment et en temps réel selon l'évolution des performances des méthodes. La décision à laquelle est attribuée la plus grande valeur de performance sera retenue sachant que la robustesse globale d'une méthode contre les fausses alarmes est calculée en moyennant les valeurs de sa robustesse face à chaque défaut.

## 5.3.2 Mise en œuvre du MIHR

## 5.3.2.1 Définition de la base de connaissances

Les sous-méthodes de diagnostic sont codées sous ROS sur des nœuds séparés envoyant leurs décisions et leurs résidus au nœud central de la MHD.

Puisque les deux algorithmes de raisonnement à partir de cas (CBR) servant à l'aide à l'isolement et à la proposition de solution possèdent des données en commun (vocabulaire dans le paragraphe 3.3), une base de données unique (de type SQL) partagée entre ces deux algorithmes est construite. Les tableaux formant cette base de données sont optimisés pour réduire le volume des données stockées comme montré à la figure 5.21.

Ainsi, trois tableaux sont définis. Le premier (indic) contient les valeurs des différents sousindicateurs de performance pour toutes les sous-méthodes. Il est relié, via un identifiant, appelé id, à un deuxième tableau (dec\_res) contenant les décisions et les résidus générés par ces méthodes. Ce tableau est relié à un troisième tableau (solutions) contenant les informations relatives aux solutions proposées.



FIGURE 5.21 – Tableaux SQL de la base de données

Les valeurs des indicateurs changent moins souvent que les autres variables. Cette architecture nous permettra d'économiser de l'espace mémoire en reliant simplement ces valeurs aux autres variables. Par conséquent, nous évitons l'insertion des données redondantes. Le temps aussi est sauvegardé dans le tableau dec\_res, il sert à retirer les n dernières lignes de données en cas de besoin.

À la figure 5.21, "rate\_m\_det" correspond au taux de détection de la sous-méthode m ; "rate\_m\_iso\_d" est le taux d'isolement de la sous-méthode m en isolant le défaut d ; "dec\_m" est la décision de la méthode m (ou de l'opérateur) ; "f\_mahal" est la distance Mahalanobis (résidu) du filtre f ; "ra\_resx" est le résidu numéro x des équations de la redondance analytique présentées dans la section 5.1.2. Le dernier tableau "solutions" contient des informations servant à la proposition des solutions. Outre les deux identifiants, ce tableau est composé de : la localisation du défaut (capteur, mécanique ou externe), la nature du défaut (dérive, biais, perte ou perturbation), l'effet ou le niveau de criticité du défaut, la solution correspondante et l'évaluation de cette solution.

#### 5.3.2.2 Définition de l'IHR

Pour communiquer avec l'opérateur, une interface graphique respectant les consignes énoncées dans la section 3.3.5 est développée.

Les informations, en relation avec le diagnostic et la supervision, sont d'abord classées dans trois groupes de priorité :

1. Priorité haute : Une information appartenant à ce groupe est une information essentielle et doit être transmise le plus rapidement possible à l'opérateur afin d'empêcher la dégradation de la situation. Nous avons choisi de montrer ce type d'informations en permanence et de les rendre lisibles plus rapidement que les autres informations par l'affichage de pictogrammes intuitifs. Par exemple, l'information sur la présence d'un défaut et l'information sur sa criticité sont des informations à priorité haute.

- 2. Priorité moyenne : Une information appartenant à ce groupe de priorité est une information utile mais n'est pas critique. Nous avons choisi de montrer ce type d'informations en permanence sans y rajouter des effets attirants (e.g., couleurs, clignotement). Par exemple, la position et la vitesse du robot sont des informations à priorité moyenne.
- 3. Priorité faible : Ne pas apprendre tout de suite une information appartenant à ce groupe de priorité ne pose pas de problème. Nous avons choisi d'afficher ces informations uniquement à la demande de l'opérateur. Par exemple, les valeurs des résidus des méthodes de diagnostic ne doivent pas être affichées tout le temps à l'écran de surveillance car ils ne sont pas utiles en cas de marche normale du robot. Toutefois, l'opérateur peut demander de voir certains de ces résidus s'il soupçonne la présence d'un défaut.

Ainsi, Les informations les plus prioritaires sont affichées de telle manière qu'elles soient perçues et comprises plus rapidement que les autres.

L'appartenance des informations aux groupes de priorité varie en fonction de la situation. La figure 5.22 montre la priorité des informations selon le mode de fonctionnement actuel.



FIGURE 5.22 – Appartenance des informations aux groupes de priorité selon le mode actuel

Conformément à la description dans le chapitre 3.3, trois modes sont considérés :

- Surveillance : Aucun défaut n'est détecté. L'opérateur surveille le robot.
- Aide à l'isolement : Un défaut est détecté sans être isolé ou bien l'opérateur a signalé un défaut. L'opérateur contribue à l'isolement du défaut.
- Aide à la proposition d'une solution : Un défaut est isolé. L'opérateur contribue à la proposition d'une solution.

Le mode surveillance est continuellement activé. L'état du robot, qui correspond à la décision prise par la MHD, y est une information hautement prioritaire; la position du robot dans la carte, la commande et la vitesse du robot sont des informations de priorité moyenne. Les résidus des méthodes de diagnostic, les mesures des capteurs et les indicateurs de performance y sont des informations à faible priorité. Quant au mode d'aide à l'isolement, les décisions des sous-méthodes de diagnostic y sont des informations à priorité moyenne et l'isolement proposé par la MHD est une information hautement prioritaire.

Finalement, dans le mode d'aide à la proposition de solution, la solution proposée par la MHD est une information de haute priorité. Les informations de la table "solutions" de la figure 5.21 y sont des informations de priorité moyenne.

**GUI** Pour répondre aux critères expliqués dans le paragraphe précédent, quatre fenêtres sont créées dont trois correspondent aux modes de fonctionnement et une correspond aux informations moins prioritaires.

La fenêtre principale, correspond au mode surveillance. Elle est toujours visible. L'état du robot y est facilement repérable grâce aux pictogrammes (normal, défaut).



FIGURE 5.23 – Interface graphique : fenêtre de surveillance

La pose, la commande et la vitesse du robot sont montrées de façon à être facilement lisibles et non distrayantes. L'évolution du robot est représentée dans une carte approximative à deux dimensions (x,y). Deux boutons y sont présents : signalement du défaut et demande d'information.

La figure 5.23 montre un prototype de la fenêtre de surveillance. Un pictogramme détermine l'état du robot, ce qui rend cette information lisible rapidement. La carte, correspondant au scénario du test, est représentée dans cette interface. L'opérateur peut choisir les informations à afficher parmi : valeurs des résidus, mesures des capteurs et performance instantanée des sousméthodes via la liste déroulante.

La fenêtre des informations permet la sélection des variables à afficher. Elles sont référencées par type de défauts que traite la méthode correspondante. Les valeurs des variables continues (qui varient fréquemment) sont tracées en temps réel (figure 5.24a) tandis que les valeurs des variables discrètes sont affichées dans un tableau (figure 5.24b).

i information	Besidual Implementing				
1 c	odometers - command		1		Rates of methods dealing with:
2 1	imu - odom	rate_c_det	0.936	1	front left wheel
	gps - odom	rate c iso gps	0.86		front sight whool
5 c	cmd - front right wheel				inonengite wheet
6 c	cmd - rear right wheel	rate_ec_det	0.924	3	3 rear left wheel
7 0	cmd - front left wheel	rate_ec_iso_gps	0.93	4	rear right wheel
9 2	gps - imu - front right wheel	rate ekf det	0.94		IA411
10 g	gps - imu - rear right wheel				
11 ç	gps - imu - front left wheel	rate_ekf_iso_gps	0.925	6	GPS
12 g	gps - imu - rear left wheel				
13 ir	imu - cmd			1	Laser
14 г	right side wheels			8	3 UWB*
15 lo	left side wheels				
16 g	gps - cmd			9	unknown
17 1	Localisation sensors				
	Localisation sensors + GPS				
40 80 80 100 120 14 19 L Pause	cuaer - rocensectori selisors				

(a) Variables continues



 $\mbox{FIGURE}$  5.24 – Interface graphique : fenêtre d'affichage des informations à la demande de l'opérateur

Quant à l'interface d'aide à l'isolement, elle est composée de trois groupes d'éléments :

- 1. Isolement proposé automatiquement : l'isolement proposé par la MHD et l'isolement proposé par l'algorithme de CBR se basant sur l'historique des décisions sont présentés à l'opérateur en lui donnant la possibilité de valider l'isolement qui lui paraît le plus logique (en cas de conflit entre les deux).
- 2. Informations importantes : conformément à la figure 5.22, les informations relatives aux décisions des sous-méthodes sont montrées. Ce sont des informations traitées de haut niveau.
- 3. Proposition d'isolement : dans le cas où l'opérateur estime que le défaut n'est pas bien isolé ni par la MHD ni par le CBR, il pourra choisir, via des listes déroulantes, l'isolement correspondant à la situation ou, en cas de doute, il pourra déclarer la nature et le type de défaut (informations qui peuvent être déduites de l'interprétation des résidus).

La figure 5.25 en montre un prototype où toutes les méthodes prédisent un mode normal. Les listes déroulantes des type "nature de défaut" sont activées si l'opérateur choisi "mode inconnu". La forme de l'interface d'aide à la proposition de solution ressemble à celle de l'aide à l'isolement. La solution proposée (par le CBR) et le défaut isolé sont montrés et maintenus visibles. La solution est colorée en vert tandis que le défaut est en rouge. La figure 5.26 illustre un prototype de l'interface de proposition de solution. L'opérateur peut valider la solution proposée, proposer une autre solution ou revoir l'isolement. Le cas présenté à la figure 5.26 représente une situation normale où aucun défaut n'est déclaré, la solution est simplement d'ignorer le défaut (ne rien faire).

😣 💷 Isolation Help									
Isolation Proposition:	м	MHD decision: Normal Valid						idate automatic isolation	
Normal ‡	Decision based on decision's history: Normal Validate isolation based on history								
Fault type Potentially useful information									
Sensor ‡		EKF2 decision	C decision	EC decision	Scan filter	UWB vector	Redundancy decision		
Fault nature	1	0	0	0	0	0	0		
Drift ‡									
Submit									

FIGURE 5.25 – Interface graphique : fenêtre d'aide à l'isolement

😣 🗊 Solution Help							
Proposed solution: Isolated fault:			Ignore Normal				
				Validate proposed solution			
Solution Proposition:	Pc	tentially useful	information:				
		Fault location	Fault nature	Effect level	Detection rate	Isolation rate	
	1	no_fault	unknown	0	0.9	0.9	
Restart 🛟							

FIGURE 5.26 – Interface graphique : fenêtre d'aide à la proposition de solution



FIGURE 5.27 – Réponse de la MHD face aux défauts

# 5.4 Résultats obtenus par l'application de S2D2R

**Résultats** Dans le but d'évaluer l'algorithme proposé, la première forme du système de supervision est mise en œuvre dans les mêmes conditions que dans lesquelles les sous-méthodes ont été testées.

Les résultats de l'application de la MHD dans les cas de chacun des défauts mécaniques et capteurs (diagnostiqués par plusieurs méthodes) sont présentés à la figure 5.27. Cette figure montre que, pour chaque défaut, la performance de la MHD est comparable à celle des meilleures des méthodes traitant le défaut. Les résultats obtenus pour les défauts d'UWB et du laser ne sont pas illustrés car ils sont exactement identiques aux sous-méthodes qui les traitent.

**Évaluation** Le tableau 5.13 synthétise la performance de la MHD calculée pour chaque défaut. Les défauts laser et UWB étant traités par une seule méthode de diagnostic, la performance de la MHD dans ces deux cas vaut celle des méthodes traitant ces défauts (EKF flexible et vecteur de confiance).

	Déte	ection	Isolement		
Défaut	Dét fav	Rob fav	Dét fav	Rob fav	
Roue avant gauche	0.946	0.963	0.914	0.898	
Roue avant droite	0.948	0.967	0.927	0.904	
Roue arrière gauche	0.947	0.965	0.927	0.905	
Roue arrière droite	0.948	0.969	0.926	0.904	
GPS	0.905	0.904	0.861	0.839	
IMU	0.958	0.974	0.889	0.854	
Laser	0.858	0.92	0.85	0.92	
UWB	0.85	0.89	0.71	0.67	

Tableau 5.13 – Évaluation de la MHD face aux défauts

Ces résultats correspondent à la forme initiale de la MHD décrite à la figure 5.20 se basant sur l'évaluation présentée dans la section 5.3.1.5. Ils ne prennent pas en compte les améliorations issues de l'interaction avec l'humain (ajustement des indicateurs de performance et refus des mauvais isolements par l'opérateur).

Pour tester le CBR et l'interaction avec l'opérateur, les étapes suivantes sont suivies :

- 1. La base de données est initiée empiriquement avec une ligne pour chaque défaut/solution.
  - La solution *stop* est choisie pour les défauts de blocage des roues;
  - La solution *intervenir* est choisie pour les défauts des capteurs de localisation (IMU et GPS);
  - Ignorer est utilisé comme solution pour le cas normal;
  - La solution *redémarrer* est associée aux défauts laser et UWB.
- 2. Le scénario de défauts de blocage de roue avant gauche est testé en premier. Le système de supervision propose automatiquement la bonne solution *stop*.
- 3. Les scenarii des défauts de blocage des autres roues sont testés ensuite. Pour chaque défaut, la bonne solution est proposée par le CBR dès le premier isolement. 10 lignes de données sont enregistrées pour chaque défaut.

4. Le scénario du défaut IMU est testé. Suite au premier isolement, le CBR propose la solution stop qui ne devait pas normalement correspondre à ce défaut. L'erreur est due à l'instabilité du CBR lié à l'insuffisance des données. En effet, la base de données contient trop de données correspondant aux blocage des roues qui, bien qu'elles ne soient pas exactement similaires à la situation de défaut GPS, en sont assez proches. Au bout de cinq itérations, nous avons refusé la solution stop et proposé à sa place la solution intervenir. Le système propose ensuite automatiquement la bonne solution intervenir suite à l'isolement du défaut GPS. À la figure 5.28, la solution intervenir est proposée suite à l'isolement du défaut IMU. La fenêtre de proposition de solutions est lancée automatiquement ainsi que celle des informations supplémentaires en montrant le résidu qui correspond à la comparaison IMU-commande. Ce résidu subit effectivement des perturbations visibles à partir du défaut.



FIGURE 5.28 – Aide à l'isolement : proposition faite par CBR

- 5. Le scénario du défaut GPS est joué ensuite. Le défaut GPS étant davantage proche du défaut IMU que d'un blocage des roues, le CBR propose correctement la solution *intervenir* pour ce défaut. 10 données sont ensuite sauvegardées dans la base de données.
- 6. Comme présenté à la figure 5.27e, des fausses détections et faux isolements apparaissent. À
chaque apparition d'un faux diagnostic (détection ou isolement) nous refusions le diagnostic et proposions le bon. Après 5 corrections pour chaque état (normal et GPS), le CBR propose directement le bon état suite au refus de la mauvaise décision. À la figure 5.29, le CBR propose le bon isolement du défaut GPS (en rouge) contrairement à l'isolement proposé par la MHD et refusé par l'opérateur.

😣 💿 Isolation Help		_	_	_	_	_	_	_	
Isolation Proposition:	MHD decision:			IMU	IMU			Validate automatic isolation	
Normal ‡	Decision based on decision's history: GPS Validate isolation based on history								
Fault type Potentially useful information									
Sensor 🌻		EKF2 decision	C decision	EC decision	Scan filter	UWB vector	undancy decis		
Fault nature	1	6	0	4	0	0	5		
Drift ‡									
Submit									

FIGURE 5.29 – Isolement du défaut IMU et proposition d'une solution

En bref, afin de se stabiliser, le CBR a besoin de  $n_{\min} = 10$  données pertinentes pour chacun des modes. Si le nombre de données constituant la base de données est modeste (< 100 points par mode), un équilibre entre le nombre de données par mode est requis, i.e., l'écart entre le nombre de données représentant un mode  $m_i$  et celui représentant  $m_j$  est inférieur au nombre de données de  $m_{min}$  (mode ayant le minimum de nombre de données dans la base). La nécessité de cet équilibre devient moins important en augmentant la taille des données appartenant à chaque mode.

Généralement, plus la base de données contient des données pertinentes, plus l'algorithme sera robuste face au déséquilibre de données et face aux erreurs humaines.

La performance des algorithmes d'aide à l'isolement et de proposition de solution (CBR) est difficile à évaluer du fait de l'évolution de la base de connaissance. Mais, à partir de 100 données par mode, le système sera robuste et les propositions pertinentes.

Quant au CBR d'aide à l'isolement, si la décision (mode) tirée par cet algorithme est différente de celle de la MHD, le mode d'aide à l'isolement est déclenché. L'opérateur aura alors les informations nécessaires pour décider du mode.

**Comparaison au cahier des charges** Le système de supervision et les méthodes de diagnostic intégrées satisfont a priori le cahier des charges défini dans 2.3.1.

En effet, ils respectent les critères suivants :

- En ligne : le système de supervision fonctionne en parallèle du robot ;
- En temps réel : le défaut est isolé rapidement après son déclenchement
- Compatibilité avec la non linéarité : les méthodes utilisées sont compatibles avec la non linéarité du modèle du robot.
- Diagnostic rapide et précis : le tableau 5.13 indique les indicateurs de détection et d'isolement correspondant à la MHD. Ces valeurs montrent que la MHD diagnostiquent les différents défauts d'une manière performante, i.e., les défauts sont détectés rapidement et avec précision.
- Diagnostic peu coûteux : comme le montre le tableau 5.4, les différentes méthodes de diagnostic contribuant à la construction de la MHD possèdent un temps de calcul par

cycle (TPC) de l'ordre de  $10^{-3}$ . Seuls les modèles de l'apprentissage profond nécessite une grande quantité de données pour l'entraînement.

- Diagnostic adaptable : Les méthodes à base de l'intelligence artificielle sont plus difficilement adaptables que les autres approches car elles exigent une nouvelle base de données pour l'entraînement. L'adaptation des autres approches est plus simple (section 5.3.1.5). L'interaction avec l'opérateur (CBR, interface graphique) est facilement adaptable 1- à l'opérateur (niveau d'expérience) et 2- à la tâche (autres applications de diagnostic).
- Diagnostic réactif : le système de diagnostic tient toujours l'opérateur au courant de la situation. L'opérateur est averti suite à l'isolement d'un défaut et, un feed-back lui informe dans le cas d'une action (application d'une solution, insertion dans la base de données).

## 5.5 Conclusion

Dans ce chapitre, la démarche de conception du système de supervision est appliquée au cas d'un robot en Skid-Steering. Les résultats de la mise en œuvre des méthodes de diagnostic sont présentés dans un premier temps. Elles sont évaluées ensuite avant d'en choisir les plus performantes pour chaque défaut afin de construire une première forme de la MHD. Les résultats de l'application des méthodes individuelles sont comparés à ceux de la première forme de la MHD. Une forme du MIHR est proposée également et la stabilité des CBRs est étudiée. Les résultats montrent, que le système proposé répond aux exigences définies dans le cahier des charges.

## Chapitre 6

## Conclusion et perspectives

Les robots mobiles jouent un rôle primordial dans le contexte de l'usine et de l'agriculture du futur. Ils interagissent entre eux et avec les humains tout en évoluant dans un environnement complexe de manière plus ou moins autonome et pour de longues tâches. Différentes architectures et différentes technologies d'automatisation/robotisation existent. Vue l'autonomie demandée à ces robots mobiles, l'apparition de défauts durant leur durée de vie est inévitable. De plus, compte tenu de la diversité des éléments constitutifs d'un robot mobile et de la variabilité de l'environnement, les sources de défauts sont très nombreuses. Il est donc nécessaire de concevoir des méthodes de diagnostic permettant de maintenir l'efficacité des robots. Ces méthodes doivent fonctionner en temps réel et doivent savoir gérer la non linéarité du système. Pour éviter les risques, le diagnostic doit être rapide et précis. En sachant que le volume de stockage et la capacité de calcul implémentés dans les contrôleurs des robots sont généralement limités, ces méthodes ne doivent pas être coûteuses en terme de temps de calcul. Finalement, le système de diagnostic doit être adaptable aux différentes conditions de fonctionnement et doit interagir avec l'opérateur d'une manière réactive afin de faciliter la reprise en main.

Les travaux décrits dans ce mémoire visent ces objectifs. D'une part, un système de supervision du diagnostic de défauts d'un robot mobile est proposé et, d'autre part, un ensemble d'outils de modélisation, de paramétrage et d'adaptation des méthodes de diagnostic aux défauts potentiels d'un robot a été mis en place.

La première partie des contributions est composée d'une méthode hybride de diagnostic mettant en œuvre plusieurs approches et types d'approche de diagnostic de défauts pour permettre le diagnostic d'une liste de défaut en temps réel. Le second est un module d'interaction homme/robot. Grâce à ce module, l'opérateur est intégré dans la boucle de diagnostic en proposant/validant des solutions mais aussi en corrigeant, si nécessaire, les décisions prises par la méthode hybride proposée. L'historique des décisions prises par l'opérateur, considéré expert, est conservée dans une base de connaissances qui est intégrée dans un algorithme de raisonnement à partir de cas afin d'améliorer la performance en continu.

La deuxième partie détaille la démarche amont et hors ligne de la mise en œuvre du système de supervision(la méthode hybride de diagnostic et le module de communication homme/robot). Elle comprend les étapes de caractérisation des défauts considérés, de définition et d'adaptation des méthodes de diagnostic capables de diagnostiquer les défauts identifiés. Cette adaptation consiste à définir les modèles, à régler les paramètres et, dans certains cas, à recueillir des données d'entraînement.

La problématique de la recherche est présentée au chapitre 2. L'hypothèse proposée y est introduite, puis elle est détaillée au chapitre 3. Le schéma global de la démarche aboutissant à la conception du système de diagnostic est énoncé et les éléments des blocs "outils de paramétrage et d'adaptation des méthodes de diagnostic aux défauts" et "système de supervision des défauts robots" sont présentés, à savoir, la démarche de la conception de la méthode hybride de diagnostic, les algorithmes de raisonnement à partir de cas et l'interface homme/robot.

Ensuite, dans le chapitre 4, les méthodes de diagnostic construisant la méthode hybride de diagnostic sont ajustées et/ou proposées pour assurer le diagnostic de la liste de défauts considérée. Les méthodes impliquées se répartissent en trois catégories : à base de modèle (EKF, UKF et redondance analytique), à base de connaissances (PFM, MMAE et RPE) et à base de données (apprentissage profond).

La démarche de conception du système de supervision est appliquée en cas d'un robot en Skid-Steering dans le chapitre 5. Les résultats de la mise en œuvre des méthodes de diagnostic sont présentés dans un premier temps. Elles sont évaluées à l'aide de l'indicateur de performance et une forme initiale de la MHD est définie grâce à cette évaluation. Une interface graphique respectant les critères définis initialement est construite et une base de données initiale est définie. Afin de se stabiliser, le CBR a besoin de  $n_{\min} = 10$  données pertinentes pour chacun des modes. Généralement, plus la base de données contient des données pertinentes plus l'algorithme est robuste. En la mettant en œuvre, la méthode hybride de diagnostic s'est montrée performante. Une forme du module d'interaction homme/robot est proposée aussi et la stabilité des algorithmes de raisonnement à partir de cas est étudiée. Les résultats montrent, que le système proposé répond aux exigences définies dans le cahier des charges.

Plusieurs pistes d'amélioration sont envisageables au niveau des méthodes de diagnostic et de leur adaptation, de l'interaction avec l'humain et de l'évaluation des algorithmes.

Au niveau des méthodes de diagnostic L'approche présentée dans 4.1.4 pour le diagnostic des perturbations du scan laser peut être étendue pour qu'il fonctionne sans avoir besoin des objets connus dans la scène. Ceci se fait en sauvegardant les objets fixes identifiés à chaque instant t = k pour les traiter comme des objets connus à l'instant t = k + 1. Les objets dynamiques représentent un défi plus difficile si leurs mouvements ne sont pas anticipés.

Étant des outils performants pour la détection des perturbations, les méthodes à base du KF peuvent être adaptées pour d'autres défauts capteurs. D'ailleurs, le PFM peut être appliqué sur d'autres défauts si nous connaissons leurs modèles. Finalement, en ayant une base de données suf-fisamment large, les modèles de l'apprentissage profond seront un outil précieux pour l'isolement des défauts.

D'autres méthodes de diagnostic et d'autres défauts peuvent être considérés et étudiés. Par exemple, il est intéressant d'étudier les cas des défauts sur les composants électroniques du robot, des défauts de suspension et des défauts de type algorithmique. Quant aux méthodes de diagnostic, elles ne sont pas toutes exploitées, comme les algorithmes d'apprentissage non supervisé et les méthodes statistiques appartenant aux méthodes à base de données.

Dans cette thèse, les résultats d'applications sont testées sur un robot de type Skid-Steering. L'adaptation de nos approches sur d'autres types de robots est plus ou moins difficile selon la méthode. Comme expliqué dans le chapitre 5, l'adaptation des méthodes à base de modèles est la plus facile. L'adaptation des méthodes à base de connaissances est un peu plus compliquée. La plus difficile reste l'adaptation des méthodes à base de données puisqu'il faudra reconstruire la base de données qui pourrait être très large et refaire l'entraînement qui est une tâche longue à accomplir.

Au niveau de l'interaction avec l'humain Le niveau d'expérience de l'opérateur n'est pas pris en compte pour l'interaction homme robot (module d'interaction homme/robot), l'opérateur est considéré comme expert dans notre étude. Il s'avère intéressant de considérer des profils différents pour les opérateurs et de dimensionner les informations échangées ainsi que les missions confiées à l'opérateur selon le profil auquel il appartient. Aussi, des améliorations sont envisageables sur l'interface graphique, nous pourrons rajouter un pictogramme pour chaque défaut pour en améliorer la visibilité, il est aussi possible d'en améliorer l'esthétique et la mise en forme et d'étudier la charge cognitive qui en résulte.

Au niveau de l'évaluation de l'approche proposée Il est intéressant de définir des scénarios type de test pour tester le système de diagnostic des défauts robots dans des contextes différents et avec des opérateurs de niveaux d'expérience variables et sur des robots différents. Puisque la performance de ce système évolue au cours du temps selon les décisions prises par l'opérateur, il faut étudier l'effet des bonnes et des mauvaises décisions sur la performance du système en fonction du nombre moyen de données dans la base de connaissances. Il est aussi utile de définir une base de connaissance générique pour chaque type de robot englobant les défauts fréquents et les solutions correspondantes.

L'autonomie des robots mobiles passe par leur capacité à s'adapter aux situations inattendues. Le système de diagnostic proposé dans cette thèse offre au robot, en coopérant avec un opérateur humain, une capacité de tolérance aux défauts. Tout comme les algorithmes de perception, de commande et de planification, il représente une brique pour la conception des robots autonomes pouvant évoluer dans des endroits incertains.

La méthodologie proposée est générique dans le sens où elle est adaptable aux différents contextes (différents robots, différents défauts et différents environnements). La méthode de diagnostic implémentée est évolutive, elle s'améliore en continu grâce à l'interaction avec l'humain.

Le système de diagnostic a un double avantage. D'un côté, il rend le robot plus résilient en détectant les défauts avant qu'ils puissent produire des pannes. D'un autre côté, il permet à l'humain d'appréhender les situations et de réagir adéquatement. En conséquence, le niveau d'acceptabilité humaine des robots devrait augmenter puisque l'humain sera en mesure de faire plus confiance au robot.

Ainsi, ces travaux contribuent à la démocratisation des robots mobiles en rendant cette technologie plus fiable. Ceci a été possible en rendant les robots mobiles tolérants aux défauts qui peuvent survenir pendant que le robot effectue des missions diverses et, en améliorant la capacité du robot à communiquer avec l'humain en cas de problème. De ce fait, les robots ainsi équipés pourront mieux servir l'humain dans les industries ou dans les champs.

## Bibliographie

- Agricultural machinery and tractors safety of highly automated agricultural machines — principles for design. ISO 18497 :2018.
- [2] Earth-moving machinery and mining autonomous and semi-autonomous machine system safety. ISO 17757 :2017.
- [3] Ergonomie de l'interaction homme-système partie 210 : Conception centrée sur l'opérateur humain pour les systèmes interactifs. ISO 9241-210 :2019.
- [4] Tractica report for agricultural robots. url : https ://www.tractica.com/newsroom/pressreleases/agricultural-robot-revenue-to-reach-74-1-billion-worldwide-by-2024/, 2019.
- [5] Naoki Akai, L.Y. Morales, Luis Yoichi, and Hiroshi Murase. Simultaneous pose and reliability estimation using convolutional neural network and Rao–Blackwellized particle filter. *Advanced Robotics*, 32(17):930–944, 2018.
- [6] Joelle Al Hage, Maan E. El Najjar, and Denis Pomorski. Multi-sensor fusion approach with fault detection and exclusion based on the Kullback–Leibler Divergence : Application on collaborative multi-robot system. *Information Fusion*, 37 :61–76, 2017.
- [7] Joelle Al Hage, Stefano Mafrica, Maan El Badaoui El Najjar, and Franck Ruffier. Informational Framework for Minimalistic Visual Odometry on Outdoor Robot. *IEEE Transactions* on Instrumentation and Measurement, 2018.
- [8] Md Alauddin, Faisal Khan, Syed Imtiaz, and Salim Ahmed. A Bibliometric Review and Analysis of Data-Driven Fault Detection and Diagnosis Methods for Process Systems. Industrial & Engineering Chemistry Research, 57(32):10719–10735, August 2018.
- [9] Alkan Alkaya. Unscented Kalman filter performance for closed-loop nonlinear state estimation : a simulation case study. *Electrical Engineering*, 96(4) :299–308, 2014.
- [10] Mahmoud Almasri, Nicolas Tricot, and Roland Lenain. Multiple model adaptive estimation for blocked wheel fault detection on mobile robots. volume 29, Warsaw, Poland, 2018. IFAC.
- [11] Mahmoud Almasri, Nicolas Tricot, and Roland Lenain. Toward an optimal assignment of diagnosis method to mobile robots faults. volume 5, Thessaloniki, Greece, 2018. IEEE.
- [12] Tiago Giacomelli Alves, Walter Fetter Lages, and Renato Ventura Bayan Henriques. Control of a mobile robot with Actuator Dynamics for Parking and Trajectory Tracking Tasks. In 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), pages 1–6, November 2018. ISSN : null.
- [13] Itamar Arel, Derek C. Rose, and Thomas P. Karnowski. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *IEEE computational intelligence* magazine, 5(4) :13–18, 2010.

- [14] Remus C. Avram, Xiaodong Zhang, and Jonathan Muse. Quadrotor sensor fault diagnosis with experimental results. *Journal of Intelligent & Robotic Systems*, 86(1):115–137, 2017.
- [15] Julia Berg, Albrecht Lottermoser, Christoph Richter, and Gunther Reinhart. Human-Robot-Interaction for mobile industrial robot teams. *Proceedia CIRP*, 79 :614–619, 2019.
- [16] Igor Bogoslavskyi and Cyrill Stachniss. Fast range image-based segmentation of sparse 3d laser scans for online operation. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 163–169, Daejeon, South Korea, October 2016. IEEE.
- [17] Soumava Boral, Sanjay Kumar Chaturvedi, and Vallayil Narayana Achutha Naikan. A case-based reasoning system for fault detection and isolation : a case study on complex gearboxes. Journal of Quality in Maintenance Engineering, 25(2):213–235, 2019.
- [18] Christophe Cariou, Benoît Thuilot, P. Martinet, and M Berducat. Guidage automatique d'un engin agricole par GPS cinématique. Journal Européen des Systèmes Automatisés (JESA), 37(3):405–433, 2003.
- [19] Jennifer Carlson and Robin R. Murphy. Reliability analysis of mobile robots. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 274–281. IEEE, 2003.
- [20] Mahalanobis Prasanta Chandra. On the generalised distance in statistics. In Proceedings of the National Institute of Sciences of India, volume 2, pages 49–55, 1936.
- [21] Bilal Daass, Denis Pomorski, and Kamel Haddadi. Using an adaptive entropy-based threshold for change detection methods-Application to fault-tolerant fusion in collaborative mobile robotics. In 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), pages 1173–1178. IEEE, 2019.
- [22] Nando De Freitas, Richard Dearden, Frank Hutter, Ruben Morales-Menendez, Jim Mutch, and David Poole. Diagnosis by a waiter and a mars explorer. *Proceedings of the IEEE*, 92(3):455–468, 2004.
- [23] Richard Dearden, Thomas Willeke, Reid Simmons, Vandi Verma, Frank Hutter, and Sebastian Thrun. Real-time fault detection and situational awareness for rovers : Report on the mars technology program task. In Aerospace Conference, 2004. Proceedings. 2004 IEEE, volume 2, pages 826–840. IEEE, 2004.
- [24] Mathieu Deremetz, Adrian Couvent, Roland Lenain, Benoit Thuilot, and Cariou Cariou. A generic control framework for mobile robots edge following. volume 2, pages 104–113, 2019.
- [25] Ali Diryag, Marko Mitic, and Zoran Miljkovic. Neural networks for prediction of robot failures. Proceedings of the Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science, 228(8) :1444–1458, June 2014. WOS :000336919200014.
- [26] Zhuohua Duan, Zixing Cai, and Huaqing Min. Robust dead reckoning system for mobile robots based on particle filter and raw range scan. Sensors, 14(9) :16532–16562, 2014.
- [27] Luigi D'Alfonso, Walter Lucia, Pietro Muraca, and Paolo Pugliese. Mobile robot localization via EKF and UKF : A comparison based on real data. *Robotics and Autonomous Systems*, 74 :122–127, 2015.
- [28] Alessandro Farinelli, Elena Zanotto, and Enrico Pagello. Advanced approaches for multirobot coordination in logistic scenarios. *Robotics and Autonomous Systems*, 90 :34–44, 2017.

- [29] Zhang Fengyun and Xu Xuesong Xuesong. Fault diagnosis of mobile robot based on variable structure multiple model algorithm. In *The 27th Chinese Control and Decision Conference* (2015 CCDC), pages 2986–2992, May 2015.
- [30] Luis J. Figueroa Rivera, Ariel Y. Ramos Ruiz, and Balasubramaniyan Chandrasekaran. A Robotic Control System for Fault Tolerance and Safety using Human Robot Interaction. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pages 0435–0439, Las Vegas, NV, USA, January 2019. IEEE.
- [31] David Filliat. Robotique Mobile. Engineering school. Robotique Mobile, ENSTA ParisTech, page 9, 2011.
- [32] Dimitrios Fragkoulis. Détection et localisation des défauts provenant des capteurs et des actionneurs : application sur un système non linéaire. page 177, 2008.
- [33] Zhou Gao, Liling Ma, and Junzheng Wang. Fault tolerant control method for displacement sensor fault of wheel-legged robot based on deep learning. In 2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA), pages 147–152. IEEE, 2018.
- [34] David Goldberg. Genetic algorithms in search, optimization, and machine learning. New York, Addison-Wesley., 1989.
- [35] Hamed Habibi, Ian M. Howard, and Silvio Simani. Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control : A review. *Renewable Energy*, pages 877–896, 2019.
- [36] Georg Helbing and Matthias Ritter. Deep Learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98 :189–198, December 2018.
- [37] Masato Hirose and Kenichi Ogawa. Honda humanoid robots development. Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences, 365(1850) :11-19, 2006.
- [38] Frank Hutter and Richard Dearden. The gaussian particle filter for diagnosis of non-linear systems. *IFAC Proceedings Volumes*, 36(5):909–914, 2003.
- [39] Rolf Isermann. Model-based fault detection and diagnosis-status and applications. IFAC Proceedings Volumes, 37(6):49–60, 2004.
- [40] Rolf Isermann. Model-based fault-detection and diagnosis-status and applications. Annual Reviews in control, 29(1):71–85, 2005.
- [41] Wei Ji and Lihui Wang. Industrial robotic machining : a review. *The International Journal of Advanced Manufacturing Technology*, 103(1-4) :1239–1255, July 2019.
- [42] Xiaohang Jin, Jicong Fan, and Tommy W. S. Chow. Fault Detection for Rolling-Element Bearings Using Multivariate Statistical Process Control Methods. *IEEE Transactions on Instrumentation and Measurement*, pages 1–9, 2018.
- [43] Marine Jouin, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, and Noureddine Zerhouni. Particle filter-based prognostics : Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, 72-73 :2–31, 2016.
- [44] Rudolph Emil Kalman and others. A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1):35–45, 1960.
- [45] Stoyan K. Kanev, Michel H.G. Verhaegen, and Gerard Nijsse. A method for the design of fault-tolerant systems in case of sensor and actuator faults. pages 837–842, 2001.

- [46] Li Ke, Tingxin Song, Nattawat Pinrath, Darren Phang Ren Yee, and Nobuto Matsuhira. Basic Experiments for a Remote Control Robot-Mapping System in Complex Environment. pages 174–179, 2019.
- [47] Eliahu Khalastchi and Meir Kalech. A sensor-based approach for fault detection and diagnosis for robotic systems. Autonomous Robots, 42(6) :1231–1248, August 2018.
- [48] H Kushwaha, Jagnnath Prasad Sinha, Tapan Khura, Dilip Kushwaha, Utpal Ekka, Mayank Purushottam, and Nishtha Singh. Status and Scope of Robotics in Agriculture. 2016.
- [49] Jean Laneurit, Roland Chapuis, and Christophe Debain. TRACKBOD, an accurate, robust and low cost system for mobile robot person following. page 6, 2016.
- [50] Roland Lenain, Mathieu Deremetz, Jean-Baptiste Braconnier, Benoit Thuilot, and Vincent Rousseau. Robust sideslip angles observer for accurate off-road path tracking control. Advanced Robotics, pages 1–15, 2017.
- [51] Roland Lenain, Johan Preynat, Benoit Thuilot, Pierre Avanzini, and Philippe Martinet. Adaptive formation control of a fleet of mobile robots : Application to autonomous field operations. pages 1241–1246, 2010.
- [52] Hongmei Liu, Dawei Liu, Chen Lu, and Xuan Wang. Fault diagnosis of hydraulic servo system using the unscented Kalman filter. Asian Journal of Control, 16(6) :1713–1725, 2014.
- [53] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234 :11–26, 2017.
- [54] Yiqi Liu and Ali M. Bazzi. A review and comparison of fault detection and diagnosis methods for squirrel-cage induction motors : State of the art. ISA transactions, 70 :400– 409, 2017.
- [55] Ying Lu, Emmanuel G. Collins Jr, and Majura F. Selekwa. Parity relation based fault detection, isolation and reconfiguration for autonomous ground vehicle localization sensors. Technical report, DTIC Document, 2004.
- [56] Matt Luckcuck, Marie Farrell, Louise Dennis, Clare Dixon, and Michael Fisher. Formal Specification and Verification of Autonomous Robotic Systems : A Survey. arXiv :1807.00048 [cs], June 2018. arXiv : 1807.00048.
- [57] Norazwan Md Nor, Che Rosmani Che Hassan, and Mohd Azlan Hussain. A review of datadriven fault detection and diagnosis methods : applications in chemical process systems. *Reviews in Chemical Engineering*, 0(0), January 2019.
- [58] 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 2018 7th International Conference on Systems and Control (ICSC), pages 148–153, Valencia, October 2018. IEEE.
- [59] Thaha Muhammed and Riaz Ahmed Shaikh. An analysis of fault detection strategies in wireless sensor networks. *Journal of Network and Computer Applications*, 78:267–287, January 2017.
- [60] Absal Nabi. IJETT Application of Least Squares Parameter Estimation Techniques in Fault Detection. International Journal of Engineering Trends and Technology - IJETT, 2013.

- [61] Kathryn B. Newhart, Ryan W. Holloway, Amanda S. Hering, and Tzahi Y. Cath. Datadriven performance analyses of wastewater treatment plants : A review. *Water Research*, 157 :498–513, June 2019.
- [62] Belkacem Ould Bouamama, Gautam Biswas, Rui Loureiro, and Rochdi Merzouki. Graphical methods for diagnosis of dynamic systems : Review. Annual Reviews in Control, 38(2) :199– 219, January 2014.
- [63] Christian Plagemann, Cyrill Stachniss, and Wolfram Burgard. Efficient failure detection for mobile robots using mixed-abstraction particle filters. In *European Robotics Symposium* 2006, pages 93–107. Springer, 2006.
- [64] Pandu Sandi Pratama, Yuhanes Dedy Setiawan, Dae Hwan Kim, Young Seok Jung, Hak Kyeong Kim, Sang Bong Kim, Sang Kwun Jeong, and Jin Il Jeong. Fault detection algorithm for automatic guided vehicle based on multiple positioning modules. In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, pages 751–757. IEEE, 2014.
- [65] Ramesh Rajagopalan. Guidance control for automatic guided vehicles employing binary camera vision. PhD Thesis, Concordia University, 1991.
- [66] Michael M. Richter. On the notion of similarity in case-based reasoning. In Proceedings of the ISSEK94 Workshop on Mathematical and Statistical Methods in Artificial Intelligence, pages 171–183. Springer, 1995.
- [67] Sanchari Sen and Anand Raghunathan. Approximate Computing for Long Short Term Memory (LSTM) Neural Networks. *IEEE Transactions on Computer-Aided Design of In*tegrated Circuits and Systems, 37(11) :2266–2276, November 2018.
- [68] Shuai Shao, Alan J. Watson, Jon C. Clare, and Pat W. Wheeler. Robustness analysis and experimental validation of a fault detection and isolation method for the modular multilevel converter. *IEEE Transactions on Power Electronics*, 31(5):3794–3805, 2016.
- [69] Zineb Simeu-Abazi and Jean-Baptiste Leger. Groupe de Recherche SPSF : Groupe de travail Maintenance.
- [70] Demetris Stavrou, Demetrios G. Eliades, Christos G. Panayiotou, and Marios M. Polycarpou. Fault detection for service mobile robots using model-based method. Autonomous Robots, 40(2):383–394, 2016.
- [71] Salvatore Strano and Mario Terzo. Review on model-based methods for on-board condition monitoring in railway vehicle dynamics. Advances in Mechanical Engineering, 11(2):168781401982679, February 2019.
- [72] Vladimír Strejc. Least squares parameter estimation. In System Identification, pages 535– 550. Elsevier, 1981.
- [73] Yu-shan Sun, Xiang-rui Ran, Yue-ming Li, Guo-cheng Zhang, and Ying-hao Zhang. Thruster fault diagnosis method based on Gaussian particle filter for autonomous underwater vehicles. *International Journal of Naval Architecture and Ocean Engineering*, 8 :243–251, 2016.
- [74] Zhao-Hui Sun, Renjun Liu, and Xinguo Ming. A Fault Diagnosis and Maintenance Decision System for Production Line Based on Human-machine Multi- Information Fusion. In Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference on ZZZ -AICCC '18, pages 151–156, Tokyo, Japan, 2018. ACM Press.

- [75] Thibault Tourrette, Roland Lenain, Raphael Rouveure, and Thomas Solatges. Tracking footprints for agricultural applications : a low cost lidar approach. page 6, 2017.
- [76] Mien Van, Shuzhi Sam Ge, and Dariusz Ceglarek. Fault estimation and accommodation for virtual sensor bias fault in image-based visual servoing using particle filter. *IEEE Transac*tions on Industrial Informatics, 14(4):1312–1322, 2017.
- [77] Vandi Verma, Geoff Gordon, Reid Simmons, and Sebastian Thrun. Real-time fault diagnosis
  [robot fault diagnosis]. IEEE Robotics & Automation Magazine, 11(2):56–66, 2004.
- [78] Vandi Verma, Sebastian Thrun, and Reid Simmons. Variable resolution particle filter. In IJCAI, pages 976–984, 2003.
- [79] Cornell Wright, Aaron Johnson, Aaron Peck, Zachary McCord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-Rivero, Ross Hatton, and Howie Choset. Design of a modular snake robot. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2609–2614, San Diego, CA, USA, October 2007. IEEE.
- [80] Sharipah Soaad Syed Yahaya, Yai-Fung Lim, Hazlina Ali, and Zurni Omar. Robust Linear Discriminant Analysis. Journal of Mathematics and Statistics, 12(4):312–316, April 2016.
- [81] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2) :229–252, December 2008.
- [82] Qing Zhao and Zhihan Xu. Design of a Novel Knowledge-Based Fault Detection and Isolation Scheme. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 34(2):1089–1095, April 2004.
- [83] Maiying Zhong, Ting Xue, and Steven X. Ding. A survey on model-based fault diagnosis for linear discrete time-varying systems. *Neurocomputing*, 306:51–60, 2018.