



HAL
open science

Diagnostic en réseau de mobiles communicants, stratégies de répartition de diagnostic en fonction de contraintes de l'application

Insaf Sassi

► **To cite this version:**

Insaf Sassi. Diagnostic en réseau de mobiles communicants, stratégies de répartition de diagnostic en fonction de contraintes de l'application. Automatique / Robotique. Université Grenoble Alpes, 2017. Français. NNT : 2017GREAT121 . tel-02128323

HAL Id: tel-02128323

<https://theses.hal.science/tel-02128323>

Submitted on 14 May 2019

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

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



THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : AUTOMATIQUE - PRODUCTIQUE

Arrêté ministériel : 25 mai 2016

Présentée par

Insaf SASSI

Thèse dirigée par **Jean Marc THIRIET**
et codirigée par **Alexia GOUIN**

préparée au sein du **Laboratoire Grenoble Images Parole Signal
Automatique**
dans **l'École Doctorale Electronique, Electrotechnique,
Automatique, Traitement du Signal (EEATS)**

Diagnostic en réseau de mobiles communicants, stratégies de répartition de diagnostic en fonction de contraintes de l'application

Thèse soutenue publiquement le **27 novembre 2017**,
devant le jury composé de :

Monsieur Christophe BERENGUER

Professeur, Institut polytechnique de Grenoble, Président

Monsieur Abdelhamid MELLOUK

Professeur, Université Paris-Est Créteil Val de Marne, Rapporteur

Monsieur Philippe WEBER

Professeur, Université de Lorraine, Rapporteur

Monsieur Antoine GRALL

Professeur, Université de Technologie de Troyes, Examineur

Monsieur Blaise CONRARD

Maître de Conférences, Université Lille 1, Examineur

Madame Alexia GOUIN

Maître de Conférences, Université Grenoble Alpes, Co-directeur de thèse

Monsieur Jean Marc THIRIET

Professeur, Université Grenoble Alpes, Directeur de thèse

Remerciements

Ce travail de thèse est le fruit de collaboration et des encouragements de plusieurs personnes que je voudrais remercier. La période de la thèse m'a permis de vivre une expérience enrichissante tant sur le plan scientifique que sur le plan humain.

Les travaux ont été dirigés par Monsieur Jean-Marc Thiriet et Madame Alexia Gouin, à qui je tiens à exprimer ma plus grande et profonde gratitude pour leur soutien, leurs conseils et pour leur disponibilité. Merci encore pour tout l'intérêt que vous avez porté à mes travaux et pour m'avoir permis de travailler en autonomie.

Je tiens à remercier Monsieur Abdelhamid Mellouk et Monsieur Philippe Weber de m'avoir fait l'honneur d'être rapporteurs de ces travaux de thèse ainsi que pour toutes leurs remarques constructives. Je remercie également Monsieur Antoine Grall et Monsieur Blaise Conrard d'avoir accepté d'être examinateurs de ces travaux et Monsieur Christophe Bérengruer de m'avoir fait l'honneur de présider la soutenance.

Un grand merci à mes collègues du laboratoire Gipsa-Lab où une belle ambiance règne. j'ai partagé ma vie professionnelle avec des gens de différents pays et de différentes cultures. Je commence à remercier mes trois premiers collègues du bureau : Raouia pour sa joie de vivre quelles que soient les circonstances, Trung pour sa sagesse et sa patience avec les trois filles du bureau et Maëlle pour sa gentillesse, son soutien et les longues discussions. Je remercie également Clément avec qui j'ai partagé des idées et qui a eu la gentillesse d'écouter mes discussions. Un grand merci pour Khaled L. et Dan de m'avoir supporté jusqu'au jour de la soutenance. Un merci également pour les collègues avec qui j'ai créé de belles amitiés. Aida, Amgad, Khaled H., Kaouther (la jeune doctorante) et Nassim (le comique), un grand merci pour votre temps et votre aide. Le jeune couple colombien Diego et Nadia pour leur humeur. François et Patricia pour les discussions culturelles enrichissantes et pour leur gentillesse. Bruno pour son soutien et son humour. Mihaly pour son soutien et ses conseils. Lucas pour son humeur, son humour et son travail rigoureux.

Merci à Ahmad Farhat, Ahmed, Ayoub, Bojan, Dacheng, Houda, Jonathan, Josué, Juan, Kazusa, Lara, Miguel, Nicolas, Quan, Rachid, Raoul, Sandra, Sebin, Soheib, Sophie, Swann, Tri, Tran, Waleed et à tous les personnels du laboratoire. Je vous remercie d'avoir rendu mes trois ans de thèse une belle période de ma vie.

Ma reconnaissance va ensuite à mes parents Alia et Hassine, mon frère Mohamed et ma sœur Afef pour leur soutien pendant mes longues années d'études et leur amour. Je n'oublie pas la grande famille, surtout mes cousines, et les amis Assia, Chayma, Faycel, Khalil, Nebil, Saber, Rawia K. et Walaa.

Table des matières

Table des sigles et acronymes	xv
1 Introduction	1
2 Systèmes commandés en réseau sans fil	7
2.1 Introduction	7
2.2 Architecture des systèmes commandés en réseau sans fil	8
2.3 Principales problématiques des systèmes commandés en réseau sans fil . . .	13
2.4 Influence du réseau sur la performance du système	16
2.5 Conclusion	25
3 Réseaux bayésiens distribués et diagnostic	27
3.1 Introduction	27
3.2 Réseaux bayésiens	28
3.3 Raisonnement avec les réseaux bayésiens	30
3.4 Inférence dans les réseaux bayésiens	31
3.5 Raisonnement distribué dans les réseaux bayésiens	34
3.6 Réseaux bayésiens pour le diagnostic	47
3.7 Réseaux bayésiens pour le diagnostic distribué	51
3.8 Conclusion	53
4 Performance et qualité de service dans les réseaux sans fil	55
4.1 Introduction	55
4.2 Contraintes temps réel des SCRs	56

4.3	Analyse de la performance de la QoS pour les systèmes commandés en réseau sans fil	57
4.4	Modèle analytique du DCF dans les réseaux IEEE 802.11	58
4.5	Définition des paramètres de la QoS	68
4.6	Cas d'études	71
4.7	Conclusion	73
5	Réseau Bayésien pour le diagnostic d'un robot mobile commandé en réseau	77
5.1	Introduction	77
5.2	Définition du système	78
5.3	Réseau bayésien pour le diagnostic : étude qualitative	85
5.4	Réseau bayésien pour le diagnostic : étude quantitative	98
5.5	Scénarios de diagnostic et d'aide à la décision	106
5.6	Reconfiguration de l'architecture de diagnostic	112
5.7	Conclusion	124
	Conclusion	127
	A Inférence dans les réseaux Bayésiens	131
A.1	Raisonnement bayésien	131
A.2	Inférence exacte dans une structure en arbre	132
A.3	Inférence avec évidence virtuelle	137
	B Plateforme de tests et de simulation	139
B.1	Présentation du robot Khepera 3	139
B.2	Architecture de commande du robot	144
B.3	Scénarios d'utilisation du banc de test	146

Table des figures

1.1	Comportement du robot mobile dans une approche de co-conception	6
2.1	Architecture d'un WNCS	9
2.2	Comparaison de la performance de chaque type de contrôle par rapport à la période d'échantillonnage [LMT02]	11
2.3	Algorithme du backoff d'accès au medium CSMA/CA [Far11]	19
2.4	Mécanisme d'accès basique du IEEE 802.11 DCF	20
2.5	Le modèle de Markov de Bianchi [Bia00]	21
2.6	Les zones de propagation du signal [Dho03]	26
3.1	Types de structures d'un RB	30
3.2	Types de raisonnement	31
3.3	Evidence virtuelle [KN03]	34
3.4	Réseau Bayésien d'un réseau de capteurs [PGM05]	35
3.5	Exemple d'un spanning tree [PGM05]	36
3.6	Échange de messages dans un spanning tree [PGM05]	36
3.7	Arbre de jonction	39
3.8	Transmission de messages via la variable d'observation dans un AEBN . .	40
3.9	Modules RBs liés virtuellement	42
3.10	Inférence par lien virtuel	43
3.11	Dépendance inter-modulaire [LLC16]	45
3.12	Méthode de liens virtuels [LLC16]	46
3.13	Relations possibles entre les différents types de nœuds du SWHM [SMM12]	50
3.14	Modèle bayésien en couches de diagnostic de [PC07]	51

4.1	Exemple d'une chaîne de Markov	59
4.2	Modèle de Markov pour la transmission des paquets de chaque boucle de commande	61
4.3	Probabilité de collision pour différentes valeurs de CW_{min}	72
4.4	Débit de saturation pour différentes valeurs de CW_{min}	73
4.5	Période d'échantillonnage critique pour différentes valeurs de CW_{min}	74
4.6	Délai moyen d'aller-retour pour différentes valeurs de CW_{min}	75
4.7	Gigue pour différentes valeurs de CW_{min}	75
5.1	Définition du système	79
5.2	Boucle de commande locale	80
5.3	Comportement du robot en simulation	82
5.4	Comparaison entre les comportements du robot en simulation et en expérimentation	83
5.5	Évitement d'obstacles	84
5.6	Comportement du robot en fonction du taux de pertes de paquets	85
5.7	Comportement du robot avec un taux de pertes de paquets de 80%	86
5.8	Comportement du robot avec un taux de pertes de paquets de 95%	87
5.9	Comportement du robot avec un retard de 2s	88
5.10	Extrait du réseau bayésien modulaire pour le diagnostic d'un robot commandé en réseau	89
5.11	Module bayésien de la partie commande	91
5.12	Module bayésien de la partie opérative	93
5.13	Module bayésien de la partie communication	95
5.14	Procédures d'initialisation et de diagnostic	108
5.15	Résultats d'inférence : scénario 1	109

5.16	Résultats d'inférence : scénario 2	110
5.17	Suppression du nœud QoS	113
5.18	Réseau Bayésien monolithique embarqué	115
5.19	Combinaison qualitative des réseaux bayésiens [HW13]	117
5.20	Extension du module bayésien de commande	119
5.21	Trajectoire du robot	120
5.22	Vitesses et orientation du robot pendant sa mission de navigation	123
A.1	Exemple d'un RB [Pea88]	133
A.2	Propagation de messages entre les nœuds	134
A.3	Exemple de propagation de messages avec une phase d'initiation	136
A.4	Exemple de propagation de messages avec une évidence virtuelle	138
B.1	Robot Khepera 3	139
B.2	Encodeur	140
B.3	Principe d'odométrie	141
B.4	Capteurs infrarouges	142
B.5	Modèle cinématique du robot	143
B.6	Évolution dU RSSI en fonction de la distance	147
B.7	Évitement d'obstacle en fonction du taux de pertes de paquets	148
B.8	Comportement du robot dans le cas d'un état <i>dégradé</i> des moteurs	149
B.9	Comportement du robot dans le cas d'un état <i>mauvais</i> des moteurs	150
B.10	Comportement du robot dans le cas d'un moteur en <i>mauvais</i> état et d'un moteur en <i>bon</i> état	151

Liste des tableaux

4.1	Notations	68
4.2	Les paramètres du réseau	71
5.1	Définition des nœuds de la partie commande	90
5.2	Définition des nœuds de la partie opérative	94
5.3	Définition des nœuds de la partie communication	97
5.4	Table de probabilités conditionnelles de l'état du contrôleur	98
5.5	Table de probabilités conditionnelles de l'état de la QoC sachant un bon état du contrôleur	100
5.6	Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur dégradé	100
5.7	Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur mauvais	100
5.8	Table de probabilités conditionnelles de l'état du robot	101
5.9	Table de probabilités conditionnelles de l'état du moteur	101
5.10	Table de probabilités conditionnelles de l'état des capteurs	101
5.11	Table de probabilités a priori de l'état de la batterie	101
5.12	Table de probabilités conditionnelles du nœud respect du délai sachant l'état du nœud probabilité de collision	102
5.13	Table de probabilités conditionnelles du nœud état du canal sachant l'état du nœud probabilité de collision	102
5.14	Table de probabilités conditionnelles du nœud pertes de données sachant l'état du nœud probabilité de collision	103
5.15	Table de probabilités conditionnelles du nœud pertes de données sachant l'état de charge du réseau	103

5.16	Table de probabilités conditionnelles du nœud respect du délai sachant l'état de charge du réseau	104
5.17	Table de probabilités conditionnelles du nœud état du canal sachant l'état de charge du réseau	104
5.18	Table de probabilités conditionnelles du nœud taux de pertes de paquets	104
5.19	Table de probabilités conditionnelles du nœud délai	104
5.20	Table de probabilités conditionnelles du nœud gigue	105
5.21	Table de probabilités conditionnelles du nœud débit	105
5.22	Table de probabilités conditionnelles du nœud connectivité	105
5.23	Table de probabilités conditionnelles du nœud RSSI sachant l'état du nœud connectivité	105
5.24	Table de probabilités conditionnelles du nœud pertes de données	106
5.25	Table de probabilités conditionnelles du nœud QoS pour une connectivité bonne	106
5.26	Table de probabilités conditionnelles du nœud QoS pour une connectivité dégradée	106
5.27	Table de probabilités conditionnelles du nœud QoS pour une connectivité mauvaise	106
5.28	Table de probabilités conditionnelles du nœud QoS pour une connectivité perdue	107
5.29	Table de probabilités conditionnelles de l'état de la QoC sachant que l'état du contrôleur est bon	121
5.30	Table de probabilités conditionnelles de l'état de la QoC sachant un état dégradé du contrôleur	121
5.31	Table de probabilités conditionnelles de l'état de la QoC sachant un état mauvais du contrôleur	121
5.32	Table de probabilités conditionnelles par apprentissage de l'état de la QoC sachant un bon état du contrôleur	122

5.33	Table de probabilités conditionnelles par apprentissage de l'état de la QoC sachant un état dégradé du contrôleur	122
5.34	Table de probabilités conditionnelles par apprentissage de l'état de la QoC sachant un état mauvais du contrôleur	122
5.35	Table de probabilités conditionnelles de la dépendance de cause-à-effet entre la QoS et la QoC	123
5.36	Table de probabilités conditionnelles de l'état de la QoC sachant un bon état du contrôleur	124
5.37	Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur dégradé	125
5.38	Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur mauvais	125
A.1	Tableau récapitulatif	137
B.1	Valeurs renvoyées par les capteurs en fonction de la distance et de la couleur de l'obstacle	143
B.2	Impact de la qualité du signal reçu sur la QoC	147

Table des sigles et acronymes

AEBN	<i>Agent Encapsulated Bayesian Network</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CW	<i>Contention Window</i>
DCF	<i>Distributed Coordination Function</i>
DIFS	<i>Distributed Inter-Frame Space</i>
EDCA	<i>Enhancement Distributed Control Access</i>
EDF	<i>Earliest Deadline First</i>
FMA	<i>Flying Machine Arena</i>
GSM	<i>Gain Scheduler Middleware</i>
HAN	<i>Home Area Network</i>
I2C	<i>Inter Integrated Circuit</i>
IAE	<i>Integral of the Absolute value of the Error</i>
ITAE	<i>Integral Time-weighted Absolute Error</i>
LPV	<i>Linear parameter-varying control</i>
MAC	<i>Medium Access Control</i>
MATI	<i>Maximum Allowable Transfer Interval</i>
MBN	<i>Modular Bayesian Network</i>
MJLS	<i>Markov jump linear system</i>
MSBN	<i>Multiply Sectioned Bayesian Network</i>
NCS	<i>Networked Control System</i>
PI	<i>Proportional Integral controller</i>
PWM	<i>Pulse Width Modulation</i>
QoC	<i>Quality of Control</i>
QoS	<i>Quality of Service</i>
RB	<i>Réseau Bayésien</i>
RSSI	<i>Received Signal Strength Indication</i>
RTS/CTS	<i>request-to-send/clear-to-send</i>
SCR	<i>Système commandé en réseau</i>
SIFS	<i>Short Inter-Frame Space</i>

SLA	<i>Service Level Agreement</i>
VoIP	<i>Voix sur IP</i>
Wlan	<i>Wireless Local Area Network</i>
WNCS	<i>Wireless Networked Control System</i>

Introduction

Les véhicules autonomes ou semi-autonomes font partie de notre vie quotidienne, nous les trouvons sur les routes, dans nos foyers pour l'aide domestique, dans l'espace, dans le domaine militaire, etc. Ces véhicules, commandés ou supervisés en réseaux, sont capables d'exécuter des tâches de nature humaine dans les milieux hostiles ou difficiles à atteindre, en particulier pour des missions de surveillance ou d'exploration. Dans ce contexte, nous pouvons mentionner les robots mobiles autonomes *Quince* [Nag+13], les robots poissons conçus par la multinationale Japonaise *Hitachi* et les robots amphibiens semi-autonomes de *Toshiba* [SK16]. Ils sont notamment intervenus en milieu dangereux, dans le réacteur *Fukushima* depuis le désastre nucléaire du 11 mars 2011. Les robots poissons sont contrôlés par des opérateurs distants pour étudier les dégâts dans une zone submergée d'eau radioactive à une profondeur de 6 mètres. Ils ont remplacé l'humain dans un milieu de très haut niveau de radiations pouvant causer la mort immédiate. Dans un autre domaine d'application, le robot mobile *Philae* de la comète *Tchouri* a servi à découvrir l'évolution du système solaire. La communication avec la station de contrôle nécessite une durée bien définie (suffisante) pour recevoir une réponse de *Philae* et lui envoyer par la suite des ordres de mission à exécuter. Il devient autonome quand la communication avec la station n'est plus disponible.

Les exemples de robotique mobile montrent que l'humain a parfois besoin de déléguer des tâches à un robot pour des missions dans des environnements dangereux, dans des lieux d'accès difficile, ou bien pour des missions de durées indéterminées. Dans toutes ces applications de la robotique mobile, le réseau de communication est un composant important du système global pour accomplir une mission bien définie. Les robots mobiles sont considérés comme des systèmes commandés en réseau sans fil (SCRs sans fil ou WNCS) dont l'utilisation se développe dans différents domaines et applications. Les composants du système sont physiquement distribués : un contrôleur implémenté sur un site distant qui envoie les commandes aux actionneurs installés sur un autre site et les capteurs qui renvoient des mesures sur l'état du système. Les actionneurs et les capteurs sont généralement implantés sur le même sous système que le contrôleur doit commander. Un réseau de communication sans fil, quatrième composant du SCR sans fil, est primordial pour assurer la communication entre le contrôleur et les actionneurs d'une part, et entre les capteurs et le contrôleur d'autre part. Les avantages de l'utilisation des réseaux sans fil dans la robotique mobile, concernent la mobilité, mais également la rapidité de déploiement et la facilité d'ajouter

de nouvelles entités dans le système ; ses propriétés entraînent l'existence de problèmes qui peuvent engendrer la dégradation de la performance du système. En effet, les SCRs sans fil sont sujets à des problèmes liés au comportement stochastique des réseaux sans fil. Ce comportement stochastique ne permet pas de garantir la transmission correcte et la bonne réception des données entre les composants du SCR : la transmission et la réception ne sont pas certaines. La mobilité du système commandé et les environnements perturbés ajoutent d'autres sources de dysfonctionnement. L'intégration du réseau sans fil dans la boucle de commande introduit des problèmes qui ont un impact sur sa performance et sa stabilité i.e, sa qualité de commande (QoC). La QoC est définie comme la capacité de garantir la stabilité, la précision, le temps de réponse, l'allure des transitoires et la robustesse d'un système. Cette QoC dépend alors de la qualité de service (QoS) qui est la capacité du réseau à fournir un service qui permet une bonne transmission des données en termes de débit, de délais de transmission, de gigue, de taux de pertes de paquets et de connectivité (un problème lié à la mobilité, à l'atténuation du signal et à la perte de connexion). Par conséquent, la performance du système va dépendre des paramètres de la QoS [SSA10]. C'est ainsi que l'étude de l'influence des défauts du réseau sans fil sur la QoC est cruciale en prenant en compte aussi les problèmes liés à la commande. On parle dans ce cas de conception réseau-commande "*co-design*", cela fait l'objet de nombreux travaux [Ber+08], [SSA10], [Mec10], [DC16], [Xu+17]. Dans cette approche, l'état du système va être amélioré en agissant soit sur la stratégie de commande, soit sur les paramètres ou l'architecture du réseau : une stratégie de "secours" qui permet d'éviter la dégradation de la mission. Cela fait des SCRs le point d'intersection de deux théories : la théorie de la commande et la théorie des réseaux de communications d'où la complexité de l'analyse de ce type de système.

En outre, les SCRs sans fil sont considérés comme des systèmes temps réel qui ont besoin d'un certain niveau de QoS pour une bonne performance. Cependant, la nature probabiliste du protocole de communication CSMA/CA utilisé dans la plupart des technologies sans fil ne garantit pas les contraintes de temps réel. Il faut alors une méthode probabiliste pour analyser et définir les exigences de l'application en terme de QoS : délai, gigue, débit, perte de paquets. Ainsi, une première contribution de cette thèse consiste à étudier les performances et la fiabilité d'un réseau sans fil IEEE 802.11 pour des SCRs sans fil qui partagent le même réseau et le même serveur de commandes [SGT17c]. À cet effet, nous avons établi un modèle probabiliste sous forme d'une chaîne de Markov qui modélise la méthode d'accès au canal de communication. Ce modèle, inspiré des deux modèles markovien [Bia00] et [TCT16], a servi pour définir le débit de saturation, le temps d'aller-retour moyen, la gigue maximale et la valeur critique de la période d'échantillonnage de la boucle de commande en réseau. Ces paramètres définissent le niveau minimal de QoS permettant de garantir une bonne QoC.

Nous travaillons, dans le cadre de la thèse, sur un véhicule mobile commandé par une station distante. Sa mission consiste à atteindre une cible en évitant les obstacles. Le réseau

sans fil ne garantit pas la transmission des données capteurs et commande. Cela provoque des problèmes qui dégradent la performance du système. Pour garantir la réussite de cette mission, une méthode de diagnostic probabiliste est primordiale à cause du comportement stochastique du système. La méthode probabiliste qui sert à surveiller le bon déroulement de la mission et l'état du robot est la deuxième contribution de cette thèse [SGT15], [SGT16]. C'est un réseau bayésien RB modulaire qui modélise les relations de dépendance cause-à-effet entre les défaillances qui ont un impact sur la performance et la stabilité du système. L'aspect stochastique et non déterministe des réseaux sans fil rend l'obtention d'un modèle analytique/mathématique, qui traduit fidèlement le comportement des SCRs sans fil, difficile. D'où l'utilité des RBs qui correspondent à un formalisme de modélisation dans l'incertain. L'utilisation du RB pour notre problématique sert à identifier la cause de la dégradation de la QoC : soit c'est un problème lié à l'état interne du robot, soit c'est un problème lié à la QoS. C'est une démarche qui permet d'éviter la dégradation de la QoC en prenant la décision qui assure la continuité de la mission. Le RB est un outil de l'intelligence artificielle qui est pertinent à utiliser dans un environnement probabiliste : l'utilisation des agents intelligents est une bonne solution pour une telle problématique [Gon+16]. Le système global a été divisé en trois sous-systèmes auxquels nous avons associé des modules bayésiens (des agents bayésiens) : un module bayésien pour le réseau, un module pour la partie commande et un autre pour la partie opérative. Cette architecture modulaire est pertinente pour un système distribué et pour faciliter la mise à jour des parties de la méthode de diagnostic. En effet, le robot ne peut être connecté à la station que si le réseau peut garantir le niveau de QoS requis. Le contrôleur est implémenté sur la station distante et le réseau intervient dans l'architecture du système et par conséquent son module RB intervient dans la procédure de diagnostic. Cependant, la station de contrôle peut interrompre la connexion avec le robot si la QoC se dégrade à cause d'une mauvaise QoS et dans ce cas le robot commute soit sur un nouveau réseau soit sur son contrôleur embarqué. Il devient autonome s'il commute sur son contrôleur embarqué et donc le réseau ne fait plus partie du système. Une politique d'adaptation de l'architecture de diagnostic est alors envisagée. La commutation d'un mode de fonctionnement débarqué à un mode de fonctionnement embarqué implique la mise à jour du RB établi. Une autre contribution consiste à définir une stratégie de commutation du mode de diagnostic : commutation d'un RB distribué à un RB monolithique embarqué [SGT17b] et vice-versa [SGT17a]. L'objectif final du travail est de définir une solution pour un problème de co-conception des SCRs mobiles sans fil. Cela consiste à trouver un compromis entre la commande et le diagnostic distribué d'une part et le réseau de communication d'autre part. La solution de reconfiguration de l'architecture de contrôle en commutant sur un contrôleur embarqué (une forme de redondance) pour garantir la sûreté de fonctionnement du robot mobile a été utilisée dans plusieurs travaux. [Lup+14] ont développé une plateforme de drones autonomes *Flying Machine Arena (FMA)* à l'université *ETH ZÜRICH*. Cette plateforme est composée d'un système de capture de mouvements, d'un réseau de communication sans fil

et d'un système de contrôle et d'estimation. Le système détermine les commandes et les envoie, via un réseau sans fil, aux drones qui doivent les exécuter pour atteindre un comportement défini à l'aide des capteurs et des calculateurs embarqués. Dans une démarche de co-conception, le pilote du drone choisit la commande appropriée en se basant sur une analyse faite par un système de détection de défaut. Ce système de détection utilise les informations envoyées par le drone sur l'état de sa batterie, les mesures de position, le taux de pertes de paquets commande et d'autres informations pour un diagnostic général. L'ensemble des composants doit être fiable et robuste tout au long de la mission du drone. La défaillance de l'un de ces composants implique le recours à un mode de fonctionnement d'urgence appelé mode à risque (*Failsafe mode*) [Lup+14]. Dans la robotique mobile, il ne faut pas attendre longtemps pour prendre une décision qui sert à sauver le système. Quand le réseau de communication ne fonctionne plus et que le contrôleur distant ne peut plus commander le drone, une première solution envisagée est de commuter sur un autre canal de communication disponible, sinon le drone commute sur un fonctionnement en mode embarqué. Si ce défaut est réglé, le fonctionnement en mode débarqué est rétabli. Un autre défaut qui peut aussi déclencher la commutation sur le contrôleur embarqué est la perte d'une hélice [MD14]. Ce contrôleur va corriger la position du drone et éviter sa chute. Cette stratégie de commutation sur des contrôleurs embarqués a été également utilisée dans le travail de [RKT17]. Dans [RKT17], le contrôleur embarqué du drone prend le relais dans le but de chasser d'autres drones malveillants. Vu les bienfaits de l'utilisation d'un contrôleur embarqué, la solution de commutation entre les contrôleurs et les réseaux, dans une approche de co-design est adoptée dans notre travail pour éviter au robot mobile de dévier de sa mission principale comme le montre la figure 1.1. En effet, si le réseau de communication est perturbé, le robot commute sur son contrôleur embarqué s'il ne trouve pas une nouvelle station de contrôle. S'il est dans la zone de couverture d'une autre station de contrôle qui peut lui garantir un bon niveau de service, le robot commute sur le contrôleur débarqué implémenté sur la nouvelle station de contrôle. Si le robot s'éloigne de la zone de couverture de la station, la puissance du signal reçu se dégrade jusqu'à la sortie de cette zone. Dans ce cas, le robot commute sur son contrôleur embarqué.

Pour étudier nos approches sur un processus réel, le robot Khepera 3, commandé par une machine via un réseau sans fil 802.11b, est utilisé pour valider notre modèle de simulation. C'est un robot mobile équipé de deux actionneurs (moteur gauche et moteur droit), de capteurs de position et de capteurs infra-rouges pour la détection d'obstacles. Il est généralement utilisé dans le milieu académique et dans la recherche pour valider des travaux dans la robotique mobile. Ce banc de tests va servir pour l'apprentissage des paramètres du RB, c'est-à-dire pour définir les probabilités marginales et conditionnelles des variables du RB. Nous implémentons la stratégie de commutation entre les contrôleurs sur Khepera 3 pour valider le comportement obtenu via les simulations. Ce mémoire de thèse est organisé en 5 chapitres de la manière suivante :

- le chapitre 2 est consacré à un état de l'art sur les SCRs, et plus spécifiquement

sur les SCRs sans fil. Nous définissons les problématiques de ce type de système et caractérisons le comportement stochastique des réseaux sans fil. Nous détaillons par la suite l'influence des paramètres de la QoS d'un réseau, dont la dynamique est stochastique, sur la QoC.

- le chapitre 3 est dédié aux RBs distribués pour le diagnostic. Nous commençons par introduire dans un premier temps les RBs, leurs propriétés et la procédure de raisonnement avec ce type d'outil. Un état de l'art sur les RBs distribués est présenté dans un deuxième temps, puis nous terminons ce chapitre avec la description et la justification de l'utilisation des RBs dans le diagnostic distribué.
- nous étudions dans le chapitre 4 la performance de la QoS dans un réseau sans fil partagé entre plusieurs boucles de commandes où nous considérons un SCR comme étant un système temps réel. Nous modélisons la procédure d'accès au canal de communication dans un réseau IEEE 802.11 à l'aide d'une chaîne de Markov. Nous définissons par la suite la période d'échantillonnage critique qui garantit une bonne QoC du point de vue réseau avec les autres paramètres de la QoS (délai aller-retour, gigue, débit). Ces paramètres vont servir à définir les seuils de performance à surveiller dans le RB pour le diagnostic du robot commandé en réseau.
- nous proposons dans le chapitre 5 une méthode de diagnostic modulaire et reconfigurable pour un SCR sans fil. Nous définissons dans premier temps l'architecture du système et nous montrons par des scénarios de simulation l'impact de la QoS sur la QoC. Nous détaillons, par la suite, la construction de notre méthode bayésienne avec des scénarios de diagnostic et de prise de décision pour commuter sur le contrôleur embarqué ou sur un autre contrôleur débarqué. Nous définissons dans un deuxième temps les procédures de commutation du diagnostic distribué modulaire au diagnostic embarqué et vice-versa. Nous présentons ensuite des scénarios de simulation pour valider l'utilisation de cette procédure et montrer l'amélioration du comportement du système. Des expérimentations ont été faites pour l'apprentissage des probabilités du RB à l'aide du banc de tests composé d'une machine et du robot Khepera 3. Une validation du comportement du robot en simulation est faite en implémentant partiellement le processus de prise de décision sur le robot Khepera 3.
- nous finissons ce manuscrit par une conclusion où nous résumons les travaux réalisés dans le cadre de cette thèse et le bilan des résultats obtenus. Un ensemble de perspectives est également présenté.

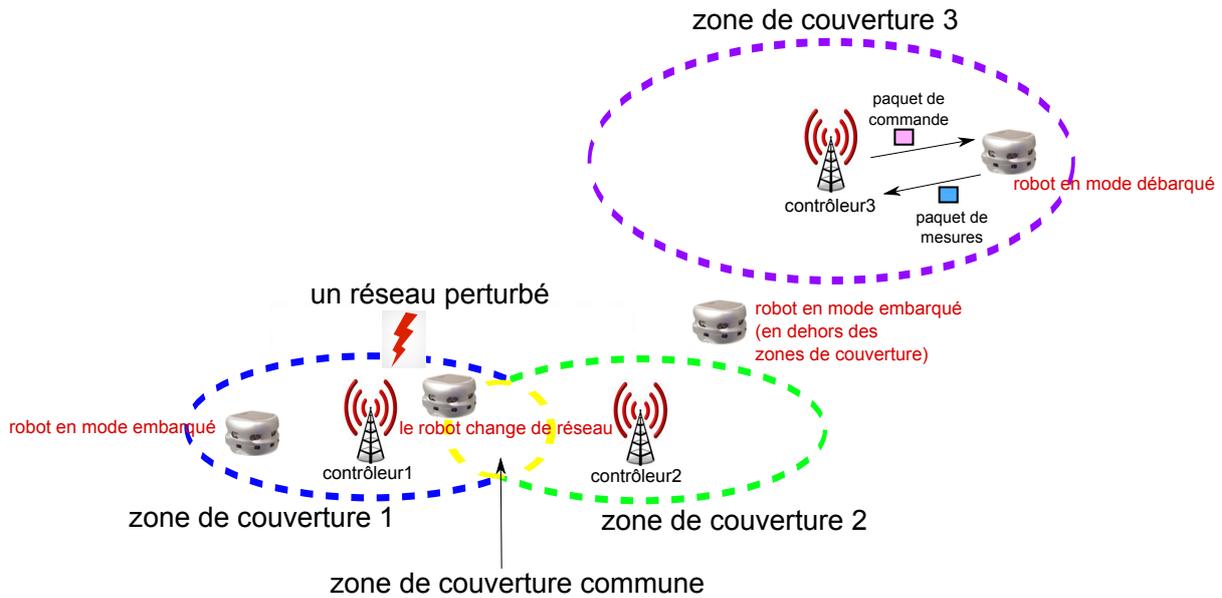


FIGURE 1.1 – Comportement du robot mobile dans une approche de co-conception

Liste de publications

Ce travail de thèse a donné lieu aux publications suivantes :

1. Sassi, I., Gouin, A., Thiriet, JM., "Diagnostic des systèmes mobiles communicants". Journées Nationales des Communications Terrestres (JNCT), Juin 2015, Valence, France.
2. Sassi, I., Gouin, A., Thiriet, JM., "A Bayesian network for diagnosis of a networked mobile robot". 26th European Safety and Reliability Conference (ESREL), Sep. 2016, Glasgow, United Kingdom.
3. Sassi, I., Gouin, A., Thiriet, JM., "Diagnosis architecture reconfiguration for a networked mobile robot". 27th European Safety and Reliability Conference (ESREL), Juin 2017, Portoroz, Slovenia.
4. Sassi, I., Gouin, A., Thiriet, JM., "Distributed to Embedded Bayesian Network for Diagnosis of a Networked Robot". The IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Juin 2017, Annecy, France.
5. Sassi, I., Gouin, A., Thiriet, JM., "Wireless Network Performance Evaluation for Networked Robots". 22nd IEEE International Conference on Emerging Technologies And Factory Automation (ETFA) , Sep. 2017, Limassol, Cyprus.

Systemes commandés en réseau sans fil

Sommaire

2.1	Introduction	7
2.2	Architecture des systèmes commandés en réseau sans fil	8
2.2.1	Architecture de contrôle via le réseau sans fil	8
2.2.2	Problématique d'échantillonnage des systèmes commandés en réseau sans fil	10
2.2.3	Stabilité : théorie de Lyapunov	13
2.3	Principales problématiques des systèmes commandés en réseau sans fil	13
2.3.1	Commande via le réseau (Control over network)	13
2.3.2	Commande du réseau (control of network)	14
2.3.3	Commande de ressources de calcul (control of the computing resources)	15
2.3.4	Co-conception des systèmes commandés en réseau sans fil	15
2.4	Influence du réseau sur la performance du système	16
2.4.1	La qualité de service	16
2.4.2	Le comportement stochastique des réseaux sans fil	18
2.4.3	Influence de la qualité de service sur la qualité de commande	23
2.5	Conclusion	25

2.1 Introduction

Les systèmes commandés en réseau SCRs (*Networked Control Systems NCSs*) ont utilisé initialement des réseaux filaires pour la transmission des informations entre les composants communicants de la boucle de commande. Les systèmes distribués et leurs applications se développent de plus en plus et font face aux limites des réseaux filaires. Pour un système de nature mobile et complexe, un réseau sans fil s'avère plus efficace pour connecter les entités communicantes. On parle alors de systèmes commandés en réseau sans fil, et l'industrie s'y intéresse de plus en plus. Cependant, l'intégration d'un réseau sans fil dans la boucle

de commande génère des problèmes qui modifient les performances du système et sa stabilité [TC03], [SSA10]. En d'autres termes, l'état du réseau et sa qualité de service (QoS) influencent la qualité de commande (QoC) du système [SSA10], [Mec10]. Comprendre le comportement des SCRs et les problématiques qui découlent de l'utilisation des réseaux sans fil est crucial. Identifier et proposer une solution pour les problèmes d'un SCR sans fil est notre objectif. Dans ce chapitre, un état de l'art sur les SCRs et leurs principales problématiques est présenté. L'influence du réseau sur la performance du système est ensuite abordée ainsi qu'une description du comportement stochastique des réseaux sans fil.

2.2 Architecture des systèmes commandés en réseau sans fil

2.2.1 Architecture de contrôle via le réseau sans fil

Les systèmes commandés en réseau sans fil (SCRs sans fil) ou Wireless Networked Control Systems (WNCSs) sont généralement considérés comme des systèmes automatiques distribués dans lesquels un réseau sans fil est intégré dans la boucle de commande comme le montre la figure 2.1. Un SCR est un système composé de trois grandes parties [Hab10], [SSA10] : la partie commande, la partie opérative et la partie communication.

- La partie commande : elle correspond au contrôleur, calculateur des lois de commande servant à piloter la partie opérative. Dans le cas d'un WNCS, le contrôleur, ayant un processeur de capacité et de puissance de calcul bien déterminées, reçoit des signaux discrets de la partie opérative et plus précisément les données des capteurs dans un "paquet de mesures". Il calcule les lois de commande puis il les envoie aux actionneurs de la partie opérative dans un "paquet de commande".
- La partie opérative : elle est composée des capteurs et des actionneurs. A chaque période d'échantillonnage, les capteurs envoient les mesures à la partie commande qui envoie, en retour, les consignes aux actionneurs via le réseau de communication. Les actionneurs, en recevant les signaux du contrôleur, exécutent les ordres de la partie commande.
- La partie communication : elle correspond au réseau de communication qui sert à l'échange d'informations entre la partie opérative et la partie commande. L'accès au réseau est défini par le protocole de communication utilisé qui dépend du type de réseau.

Les informations échangées entre les différents composants sont des signaux à temps discrets envoyés dans des paquets. La conception des SCRs nécessite des compétences dans trois domaines : les réseaux de communications, la théorie de la commande et l'informatique temps réel. En effet, le réseau de communication a un comportement dynamique qui doit

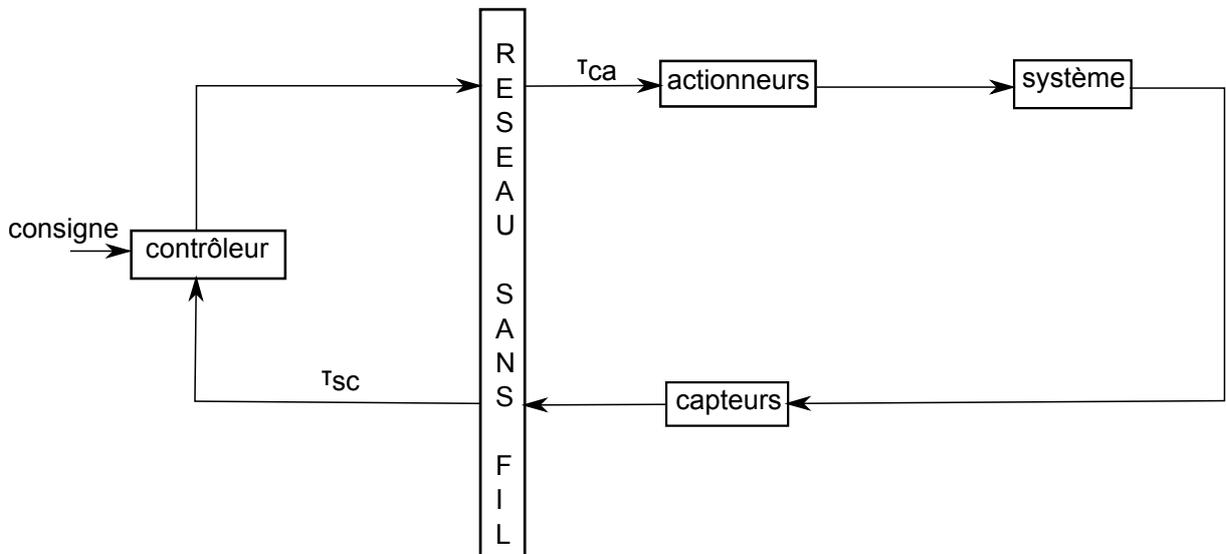


FIGURE 2.1 – Architecture d'un WNCs

être pris en considération dans la conception des SCRs. Les retards de transmission, la gigue (variation du retard), les pertes de paquets, la désynchronisation d'horloges et d'autres paramètres de la QoS et du réseau peuvent dégrader la performance de la boucle de contrôle et déstabiliser le système [SSA10], [PH16]. Par conséquent, le WNCs est un système temps réel qui nécessite un certain niveau de QoS. Dans un réseau de WNCs, un nœud capteur envoie périodiquement un paquet de mesures au contrôleur qui contient les mesures de l'état. Quand le contrôleur reçoit les données capteurs, il commence le calcul des lois de commande. Il crée ensuite un paquet de commande et l'envoie aux actionneurs du système à commander. Les deux paquets sont envoyés via le réseau sans fil ce qui va engendrer un retard dans la boucle de commande comme le montre la figure 2.1. Les deux retards qui apparaissent dans la boucle de commande sont le retard capteur-contrôleur τ_{sc} et le retard contrôleur-actionneur τ_{ca} [TC03]. Le retard τ_{sc} est le retard constaté à la réception du paquet de mesures et le retard τ_{ca} est le retard à la réception du paquet de commande. Le WNCs est un système temps réel qui doit respecter la contrainte temps réel suivante [TC03], [TCT16] :

$$T_{RTT} \leq T \quad (2.1)$$

où T_{RTT} le temps d'aller retour tel que $T_{RTT} = \tau_{sc} + \tau_{ca}$ et T est la période d'échantillonnage de la boucle de commande. Ces délais ne sont pas constants et sont aléatoires vu le comportement stochastique des réseaux sans fil. La QoC est un paramètre d'évaluation de la performance de la boucle de commande et qui dépend de la QoS, de l'algorithme de commande et de la période d'échantillonnage.

2.2.2 Problématique d'échantillonnage des systèmes commandés en réseau sans fil

La technologie analogique disparaît face à la croissance des systèmes calculateurs numériques à grande puissance et à faible coût depuis quelques années [ÅW97],[SSA10]. L'utilisation des contrôleurs numériques est répandue dans plusieurs applications induisant des problématiques d'échantillonnage et de désynchronisation. Les informations traitées par le calculateur numérique proviennent des signaux continus de capteurs qui sont échantillonnés. La théorie d'échantillonnage doit être prise en compte, par conséquent, dans l'étape de conception du système. Avec le déploiement des SCRs, qui provoquent la dispersion des capteurs, des actionneurs et des contrôleurs, d'autres contraintes liées au réseau de communication se rajoutent dans la conception du système. Le comportement non fiable des réseaux sans fil engendre des retards et des pertes de paquets et crée des problèmes de timing. Ainsi, la période d'échantillonnage d'une boucle de commande implémentée via le réseau dépend des retards et de la disponibilité du réseau. La performance du SCR doit être étudiée au cours de sa conception. Le choix de la période d'échantillonnage doit être intégré à cette étape de conception. La figure 2.2 de [LMT02] montre comment définir la bonne période. Les signaux envoyés via le réseau sont des signaux discrets comme cela a été indiqué dans la définition de l'architecture de contrôle via le réseau. En effet, il existe deux méthodes d'échantillonnage : un échantillonnage temporel qui est périodique (time-triggered sampling) et un échantillonnage événementiel (event triggered sampling). L'échantillonnage événementiel est une stratégie qui consiste à mettre à jour la commande seulement dans le cas où un événement apparaît [MDC13]. Un événement est généré à partir d'une fonction d'événements qui définit l'instant où il faut mettre à jour la valeur de la commande. Il faut évaluer cette fonction périodiquement pour déterminer l'état du système. Une fonction d'événements e est définie telle que $e : X \times X \rightarrow \mathbb{R}$, si $e \leq 0$, la commande doit être mise à jour, où X représente l'ensemble de variables d'état. L'échantillonnage périodique, simple à mettre en œuvre, est souvent utilisé dans les systèmes numériques [ÅW97], [ZHY16]. Pour les systèmes commandés en réseau, la stabilité de la boucle de commande dépend de la période d'échantillonnage et du retard capteur-actionneur $\tau_{sc} + \tau_{ca}$. Il faut donc respecter un timing adéquat pour avoir une bonne performance et une bonne QoC. Le choix de la période d'échantillonnage est une tâche primordiale et un compromis entre la performance de la boucle de commande et les ressources utilisées [SSA10]. [LMT02] a étudié la performance de trois types de commandes en fonction de la période d'échantillonnage comme le montre la figure 2.2. Elle est alors considérée comme inacceptable, acceptable en se basant sur l'erreur statique des variables d'états, le dépassement, la stabilité. Pour la commande continue (systèmes analogiques), la performance est constante pour une loi de commande définie et ne dépend pas de la période échantillonnage. Pour la commande numérique, la performance du système est fonction de la période d'échantillonnage. Le point A correspond à la limite de la performance de la commande numérique, et la période d'échantillonnage

correspondante est déterminée à partir de la bande passante du système. En effet, cette valeur critique doit être choisie dans une marge de périodes d'échantillonnage définie en se basant sur le théorème de Shannon. Cette marge est établie selon la dynamique et la bande passante désirée de la boucle de commande fermée. D'après le théorème de Shannon, il faut prendre au moins deux échantillons par période de signal T_s , c'est-à-dire $2 \times T_e \leq T_s$, T_e représentant la période d'échantillonnage. La règle pour satisfaire la condition de Shannon et avoir "une bonne marge de sécurité" est la suivante : $\frac{T_s}{26} \leq T_e \leq \frac{T_s}{5}$. [ÅW97] a également défini une marge en fonction du temps de montée t_m de la réponse du système, autrement dit le nombre d'échantillons par temps de montée :

$$4 \leq \frac{t_m}{T_e} \leq 10 \quad (2.2)$$

Ces règles de choix ne donnent qu'une information de la période d'échantillonnage préliminaire à implémenter. La valeur qui va être choisie doit être ajustée par des simulations et des expérimentations. La figure 2.2 montre que la performance du système commandé en

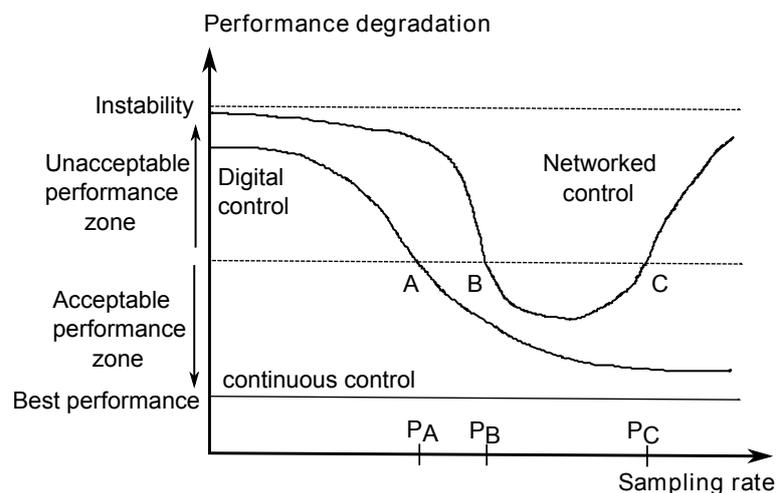


FIGURE 2.2 – Comparaison de la performance de chaque type de contrôle par rapport à la période d'échantillonnage [LMT02]

réseau est acceptable pour une période d'échantillonnage comprise entre P_B et P_C . Le point B , qui correspond à la plus grande période d'échantillonnage acceptable, est déterminé en étudiant la bande passante du système et la réponse du système. Le point C est déterminé en étudiant les caractéristiques et en analysant les statistiques de la charge réseau. Pour des petites périodes d'échantillonnage, la performance d'un système de commande échantillonné s'approche de celle d'un système continu. En revanche, pour les SCRs, le réseau de communication va être surchargé ce qui va provoquer des retards dans le réseau et par conséquent une dégradation de la QoS [ZHY16]. De plus, il y aura une surcharge au niveau du calculateur des lois de commande (serveur des lois de commande). La dégradation de la QoS et la surcharge du processeur du calculateur vont dégrader la QoC du système.

[LMT02] ont défini la plus petite période d'échantillonnage P_C tel que : $P_C = \frac{T_{ttt}}{U}$ où T_{ttt} représente le temps de transmission total et $U = \frac{t_d}{t_{overhead}}$ est le ratio maximum d'utilisation du réseau. Le temps t_d est le délai de transmission des données utiles et $t_{overhead}$ est le délai de transmission des données supplémentaires comme les entêtes des trames, l'acquittement et d'autres délais qui correspondent au protocole de communication utilisé.

De manière similaire, [CIS07] ont défini la plus petite période d'échantillonnage h_{min} pour maximiser la bande passante du canal de communication dans un réseau 802.11b tel que : $h_{min} = \frac{taille_{toutes\ les\ donnees}}{S}$ où S est le débit maximum effectif atteint ou le débit de saturation. L'expression de P_C dans [LMT02], ou de h_{min} dans [CIS07] ne reflète pas le comportement stochastique des réseaux CSMA/CA. Dans le chapitre 4, une méthode stochastique sera présentée pour déterminer les différents paramètres liés à la QoS, en particulier le débit de saturation S . Cette méthode prend en compte le comportement probabiliste dans un réseau CSMA/CA.

[HNX07] ont décrit la procédure de détermination de la valeur maximale acceptable de la période d'échantillonnage h pour les SCRs linéaires et les SCRs non linéaires. Cette valeur maximale, appelée *maximum allowable transfer interval (MATI)* est telle que $h \in [h_{min}, h_{max}]$ avec $h = t_{k+1} - t_k$, h_{max} la plus grande période d'échantillonnage, t_k et t_{k+1} les instants d'échantillonnage. Cette analyse sera reprise ultérieurement dans le chapitre 4 pour définir les intervalles des périodes d'échantillonnage adéquats pour notre système commandé en réseau. Dans le but d'optimiser l'utilisation des ressources existantes dans le réseau et dans le processeur de calcul de commande, une politique d'adaptabilité de la période d'échantillonnage a fait l'objet de plusieurs travaux de recherche comme [SSA10], [CIS07]. [CIS07] ont proposé une méthode pour adapter la période d'échantillonnage dans un réseau 802.11b en tenant compte des conditions du réseau et de l'état de la QoS et plus précisément du temps d'aller retour. Ce temps consiste en un retard capteur-contrôleur et un retard contrôleur-actionneur. Par conséquent, une période d'échantillonnage sera attribuée à chaque état du retard (plus précisément pour chaque intervalle de valeurs du retard). L'échantillonnage non uniforme pose des problèmes de stabilité. En effet, le switching entre les contrôleurs, ayant chacun une période d'échantillonnage, sans suivre une bonne stratégie crée une instabilité [SSA10]. [CIS07] propose une conception du WNCS dans un environnement probabiliste en utilisant une chaîne de Markov le "*Markov jump linear system*" (MJLS) pour étudier la stabilité stochastique du système pour chaque variation de la période d'échantillonnage. Dans d'autres travaux, l'approche Linear Parameter Varying (LPV) a été utilisée pour renforcer la stabilité pendant le switching [SSA10]. C'est une approche qui se base sur un ordonnancement de gain avec un contrôleur robuste LPV garantissant une stabilité quelle que soit la variation de la période d'échantillonnage d'après [SSA10]. Vu l'aspect stochastique des réseaux sans fil, les prises de décision pour faire varier la période d'échantillonnage sont établies à l'aide d'un framework probabiliste dans notre travail.

2.2.3 Stabilité : théorie de Lyapunov

La QoC est la capacité de garantir la stabilité, la précision, le temps de réponse, l'allure des transitoires et la robustesse des systèmes. Afin d'étudier la stabilité d'un WNCS, qui est un paramètre important de la QoC, la théorie de Lyapunov est utilisée. La stabilité du WNCS, qu'on peut considérer comme un système à retard, est analysée avec la seconde méthode de Lyapunov. Cette méthode consiste à utiliser une fonction $V = V(t, x(t))$ définie positive sur l'espace d'état d'un système telle que sa dérivée $\frac{dV}{dt}$ est définie négative. Cette fonction dépend de l'état actuel de la variable d'état x . Pour les systèmes à retard, il va falloir prendre en compte l'état actuel de la variable d'état et ses valeurs passées. Par conséquent, deux autres extensions de la seconde méthode de Lyapunov ont été établies : Lyapunov-Razumikhin et Lyapunov-Krasovski [SSA10]. La première extension consiste à vérifier la deuxième condition de stabilité de la seconde méthode de Lyapunov que dans l'intervalle de retard. En effet, il faut que la dérivée de $V = V(t, x(t))$ soit inférieure ou égale à 0 pour tout instant $s \in [t - \tau, t]$ avec τ est un réel strictement positif. Quant à Lyapunov-Krasovski, cette méthode consiste à trouver une fonctionnelle qui soit décroissante dans toutes les trajectoires.

2.3 Principales problématiques des systèmes commandés en réseau sans fil

L'utilisation des réseaux sans fil et des calculateurs numériques dans la boucle de commande ajoute des contraintes dans l'étude de la performance d'un WNCS. Ainsi, la QoS, l'algorithme de commande, les ressources de calcul (processeurs et mémoires) et la QoC doivent tous faire partie de la conception du WNCS. Pour analyser les performances d'un tel système, l'étude du WNCS doit alors être globale : on applique une approche de co-design [SSA10]. En outre, il existe deux autres approches pour analyser les problèmes des SCRs [GC10]. La première approche appelée *control of network* consiste à concevoir un SCR qui ne dépend pas des problèmes de la QoS (délais, pertes de paquets...) et qui fournit un support de communication qui minimise l'occurrence de ces problèmes. La deuxième approche appelée *control over network* consiste à concevoir un algorithme de commande robuste ou tolérant aux défaillances du réseau.

2.3.1 Commande via le réseau (Control over network)

L'approche de commande via le réseau (*control over network*) traite de la conception d'un SCR qui prend en compte les paramètres de la QoS. En effet, la QoS peut se dégra-

der à cause d'une congestion réseau ou de la dégradation de la qualité du signal. Il faut donc garantir une bonne QoC quelles que soient les conditions réseaux. Pour minimiser les impacts du réseau sur la QoC, le concepteur d'un SCR utilise soit un algorithme de commande qui modélise la défaillance réseau (le retard par exemple) soit un algorithme qui compense l'effet de cette défaillance [GC10], [SSA10]. Par exemple, [TC03] ont développé une procédure de mise à jour de gain du contrôleur *Gain Scheduler Middleware* (GSM) pour compenser l'effet du délai sur la boucle de commande.

2.3.2 Commande du réseau (control of network)

La méthodologie de commande du réseau (*control of network*) consiste à concevoir un réseau qui garantit la QoS nécessaire à l'application, c'est-à-dire une QoS qui permet de garantir les contraintes temps réel et la stabilité du système. De cette approche, il faut gérer les ressources du réseau en terme de bande passante et de priorité pour minimiser le retard et les pertes de paquets [SSA10]. La commande du réseau consiste à utiliser des mécanismes d'accès au médium qui garantissent le déterminisme de la transmission ou des méthodes d'ordonnancement. Pour les réseaux sans fil, le standard IEEE 802.11e EDCA [Man+03], [Mal+15] a rajouté des mécanismes pour garantir un bon fonctionnement des applications temps réel en donnant une priorité à chaque classe de trafic et en attribuant la priorité la plus haute aux trafics temps réel. [Zha+08] ont proposé le 802.11 *EDCA+* une version modifiée de 802.11 *EDCA* en configurant des paramètres du protocole d'accès comme la valeur minimale de la fenêtre de contention et le nombre limite de retransmissions pour améliorer les paramètres de la QoS. [TCT16] ont amélioré le mécanisme de backoff dans la couche MAC en proposant un algorithme (*Contention Sensitive Backoff Algorithm*) qui double la taille de la fenêtre de contention non pas dans le cas d'une collision seulement mais aussi si le canal est occupé. [TCT16] ont montré la fiabilité de leur méthode avec une modélisation markovienne pour étudier la performance de la QoS avec ce nouvel algorithme. Pour les méthodes d'ordonnancement dans les files d'attente, des algorithmes d'ordonnancement ont été utilisés tels que EDF et le (m, k) firm [SSA10].

Pour définir le niveau requis de la QoS qui garantit la QoC, une étude analytique est faite dans le chapitre 4. Une chaîne de Markov est établie pour déterminer les paramètres de la QoS qui définissent un comportement déterministe d'un réseau DCF quand plusieurs boucles de commande partagent le même réseau.

2.3. Principales problématiques des systèmes commandés en réseau sans fil 15

2.3.3 Commande de ressources de calcul (control of the computing resources)

La théorie du contrôle et les calculateurs temps réel ont fait l'objet d'études pour les systèmes industriels, les systèmes mobiles... Les calculateurs temps réel doivent être fiables et disponibles pour fournir aux systèmes leurs valeurs de commande. Un calculateur, comme le réseau de communication, peut être partagé entre différentes tâches à exécuter. La charge du processeur de calcul de lois de commande peut impacter la QoC. Il faut donc trouver une méthode pour gérer la répartition de l'utilisation du processeur entre les tâches qui le partagent. Les méthodes d'ordonnancement proposent une solution pour mieux gérer l'accès au processeur de calcul [SSA10]. Une méthode d'ordonnancement *Feedback scheduling* a également permis de réguler les paramètres d'ordonnancement en fonction de la QoC en établissant une boucle de contrôle pour les ressources d'un CPU (*scheduling regulator*) [SSA10].

2.3.4 Co-conception des systèmes commandés en réseau sans fil

L'approche de co-design consiste à étudier simultanément la dynamique du réseau en terme de QoS et celle de la commande en analysant sa QoC. Elle consiste à étudier les algorithmes d'accès au médium et leurs stratégies d'ordonnancement et la stabilité des systèmes. [Mec10] ont proposé une méthode pour adapter la stratégie de contrôle des robots mobiles par une station de contrôle en vue de maintenir une bonne QoS en ordonnant à un des robots à commuter sur un mode de fonctionnement embarqué. Cela consiste à abandonner l'architecture de contrôle implémentée via un réseau et à utiliser une boucle de commande embarquée dans le robot dont les performances sont dégradées. Pour garantir une bonne QoC, ils ont ensuite proposé une procédure de *handoff* qui permet de changer de réseau et de station de contrôle pour améliorer la QoS et pour avoir davantage de ressources de calcul dans le nouveau calculateur de contrôle. [Xu+17] ont développé une stratégie de contrôle tolérante aux fautes pour un WNCS décentralisé, composé de N boucles de commandes, en tenant compte de l'effet de l'ordonnancement. [SSA10] ont proposé un modèle de serveur de commande qui doit exécuter une tâche de commande pendant une durée de temps bien définie en utilisant un algorithme d'ordonnancement bien déterminé. [DC16] ont développé une méthode de conception qui consiste à établir un algorithme de commande optimale pour un réseau de communication donné et puis un réseau optimal pour une stratégie de commande. C'est une méthode itérative qui commence par une structure initiale de contrôle pour un réseau caractérisé par certaines conditions puis en régulant les variables du contrôleur, l'algorithme converge vers une structure de contrôle bien définie. [CIS07] ont établi un modèle markovien (modèle stochastique) pour mettre à jour la période d'échantillonnage de la boucle d'un WNCS en se basant sur la

valeur du retard du réseau. Ils ont aussi défini la plus petite période d'échantillonnage que le réseau peut supporter.

2.4 Influence du réseau sur la performance du système

Les systèmes commandés en réseau sans fil, comme les robots mobiles communiquant avec une station distante, sont une nouvelle génération de systèmes distribués où la communication entre les actionneurs, les capteurs et le contrôleur est assurée par un réseau de communication sans fil comme cela a été décrit dans l'architecture des SCRs de la section 2.2. La QoS et ses paramètres ont une influence importante sur la stabilité d'un WNCS et sur sa QoC. Cette partie est consacrée à l'étude de cette influence afin de montrer les motivations de notre contribution.

2.4.1 La qualité de service

La qualité de service QoS désigne la capacité du support de communication à garantir des exigences en matière de temps, de bande passante et un niveau acceptable de pertes de paquets. Chaque application a besoin d'un niveau bien défini de QoS pour lui garantir une bonne performance. Il existe deux types de trafic [Cen+07] :

- **Trafic Best-effort** : ce type de trafic correspond aux applications dont la transmission de données n'est pas sensible à la QoS i.e, au temps. Il faut juste garantir une transmission et une réception correcte de paquets. E-mail, navigateur web, Telnet, FTP sont des exemples d'applications best-effort. Ce trafic possède la priorité la moins importante. [Cen+07]
- **Trafic temps réel** : c'est un trafic sensible à la QoS et ses métriques. Les applications correspondant à ce type de trafic ont besoin de recevoir leurs données pendant un temps bien défini pour un délai et une gigue bien définis. Les paquets temps réel ne sont pas retransmis ou bien il existe une limite de retransmission afin de garantir une transmission à court délai. Les applications temps réel qui génèrent ce type de trafic sont les voix sur IP (VoIP), les vidéos conférences, etc... Les WNCSs, considérés comme des applications temps réel périodiques, ont deux types de paquets (les paquets de mesures et les paquets de commande) qui doivent être reçus avant leurs dates limites. De plus, la perte de paquets est tolérée sous un certain seuil. Afin de répondre à ces exigences en terme de QoS, qui seront définies dans le chapitre 4, les WNCSs ont tendance à faire fonctionner leurs nœuds dans leurs capacités maximales pour satisfaire les contraintes de temps réel. En outre, la priorité la plus importante est consacrée au trafic d'alarme/ de notification qui est un trafic de sécurité critique (*safety traffic*) comme les messages *Goose*.

2.4.1.1 Délai de bout en bout

Les applications temps réel ont besoin de délivrer un paquet dans un délai bien défini. Le délai de bout en bout est un paramètre très important de la QoS et a un impact sur la QoC. Il mesure le retard introduit dans la transmission des paquets dû à la charge du réseau et à la qualité du canal de transmission. Le retard se compose :

- du délai de traitement $d_{traitement}$: ce délai désigne le temps nécessaire pour traiter les informations du côté de la station émettrice.
- du délai de transmission $d_{transmission}$: c'est le délai entre le début de transmission du premier bit et la fin de transmission du dernier bit du paquet. Ce délai dépend de la taille du paquet P transmis et du débit du réseau $debit$ tel que $d_{transmission} = \frac{P}{debit}$
- du délai de propagation $d_{propagation}$: c'est le temps nécessaire au signal pour parcourir le chemin entre la station émettrice et la destination et qui dépend de la distance qui les sépare tel que : $d_{propagation} = \frac{d}{v}$ où d est la distance entre la source et la destination et v correspond à la vitesse de propagation du signal dans le canal de communication.

Les délais τ_{sc} et τ_{ca} , qui sont représentés sur la figure 2.1, sont composés de ces délais. Ils dépendent finalement de la bande passante maximale, de la taille des paquets et des spécifications du protocole de communication.

2.4.1.2 Gigue

La gigue est un paramètre très important pour les applications temps réel comme les WNCSs. Elle désigne la variation du retard de bout en bout due aux conditions du réseau. C'est la différence de délai de transmission de bout en bout entre des paquets choisis dans un même flux de paquets.

2.4.1.3 Le taux de pertes de paquets

Le taux de perte de paquets est un paramètre important de la QoS. C'est le pourcentage de paquets perdus et qui n'arrivent pas à leur destination pendant un laps de temps. La perte de paquets peut être causée par la charge/congestion du réseau, la perte de la connectivité entre les entités communicantes, la dégradation de la puissance du signal et un long délai pour les applications temps réels. Une retransmission est envisageable pour assurer la réception des paquets mais dans le cas des applications temps réel, le nombre de retransmissions doit être limité.

2.4.1.4 Débit

Le débit désigne la quantité de données transmises par unité de temps i.e., le nombre de bits transmis par seconde. Il représente la bande passante allouée pour chaque application interférant avec le réseau. Le débit alloué à une application, considéré comme une ressource qui lui est dédiée (selon les besoins de l'application), dépend de la taille de la trame à envoyer et du réseau. Cette limite en bande passante peut entraîner des défauts au niveau du système commandé en réseau.

2.4.2 Le comportement stochastique des réseaux sans fil

L'utilisation des réseaux sans fil présente plusieurs avantages que [Li+15] ont décrits : la flexibilité, l'absence de câblage, la mobilité et le déploiement rapide. Par conséquent, ils ont des caractéristiques qui conviennent au fonctionnement d'un WNCS mobile (des robots mobiles commandés par une station de contrôle). Ce type de réseau possède une architecture qui permet de gérer dynamiquement le nombre de nœuds communicants. Les réseaux locaux sans fil (Wireless local area networks (Wlan)) du standard IEEE 802.11 sont utilisés dans plusieurs applications à large zone de couverture comme les WNCS. Cependant, leur manque de fiabilité peut engendrer plusieurs problèmes. Le comportement stochastique ne garantit pas la transmission et la réception des données. En effet, la probabilité d'une transmission réussie dépend de plusieurs facteurs liés aux conditions environnementales, à la qualité du signal et aux collisions qui peuvent se produire à cause du nombre de nœuds qui partagent le même canal de communication. Pour gérer l'accès au médium de plusieurs entités communicantes, la couche MAC du standard IEEE 802.11 est basée sur un mécanisme d'accès aléatoire appelé Distributed Coordination Function (DCF). C'est un mécanisme aléatoire caractérisé par un mode d'accès appelé Carrier Sense Multiple Access With Collision Avoidance CSMA/CA. Le comportement stochastique du DCF doit être analysé afin de définir la QoS minimale permettant de garantir une bonne QoC.

2.4.2.1 IEEE 802.11 DCF

Pour les réseaux sans fil, le DCF se base sur la technique du CSMA/CA [Man+03]. Il existe deux modes d'accès : le mode basique et le mode request-to-send/clear-to-send (RTS/CTS). Le mécanisme CSMA/CA est conçu pour réduire les collisions dans un canal de communication partagé entre des stations qui tentent d'émettre simultanément. Chaque station ayant un paquet à transmettre écoute le canal pour vérifier s'il est occupé ou libre. Dans le cas où le canal est libre pendant une période de temps appelée *Distributed Inter Frame Space* (DIFS), la station transmet son paquet. Dans le cas contraire, la transmission

est différée jusqu'à ce que le canal soit libre pour une période de temps égale à DIFS comme le montre la figure 2.3 de l'algorithme de backoff. Après cette période, la station initialise

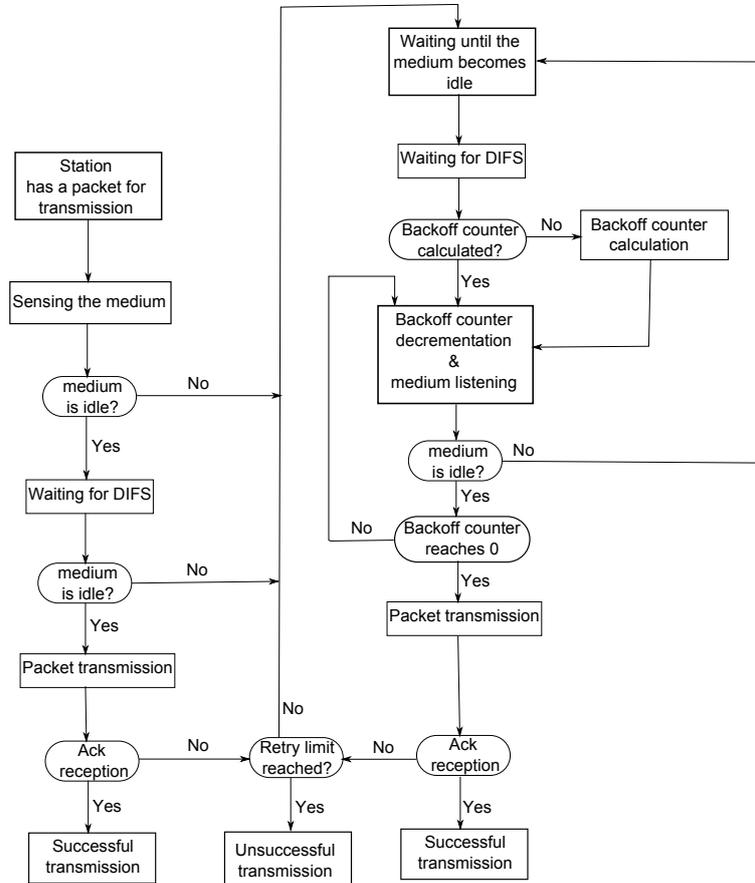


FIGURE 2.3 – Algorithme du backoff d'accès au medium CSMA/CA [Far11]

le temps du backoff avec une valeur aléatoire choisie dans un intervalle de la taille de la fenêtre de contention. La station décrémente ce temps tant que le canal est libre. Quand une transmission est détectée, la décrémentation du temps de backoff s'arrête et ne sera reprise que lorsque le réseau deviendra libre pendant une période DIFS comme le montre la figure 2.4. Quand le temps de backoff arrive à zéro, la station peut commencer la transmission de son paquet. A chaque transmission de paquet, le temps de backoff est choisi uniformément dans l'intervalle $[0, CW - 1]$ où CW est la taille de la fenêtre de contention. Pour la première tentative de transmission, CW est égale à CW_{min} . Après chaque échec de transmission, CW est doublée jusqu'à une valeur maximale CW_{max} telle que :

$$\begin{aligned} CW_j &= 2^j \times CW_{min} \\ CW_{max} &= 2^R \times CW_{min} \end{aligned} \quad (2.3)$$

avec $j \in [0..R]$ et R le nombre maximal de retransmissions.

Quand CW atteint la valeur maximale CW_{max} , elle doit garder cette valeur jusqu'à ce

que le paquet soit transmis avec succès ou que le compteur de retransmission ait atteint la valeur limite de retransmission. Quand la limite de retransmission est atteinte, la station abandonne la transmission du paquet. Pour savoir si la réception des données est réussie ou s'il y a eu une collision, la station émettrice attend un acquittement de réception (paquet Ack) transmis par la station destination après une période de temps appelée *Short Interframe Space* (SIFS) comme le montre la figure 2.4. Le canal de communication reste

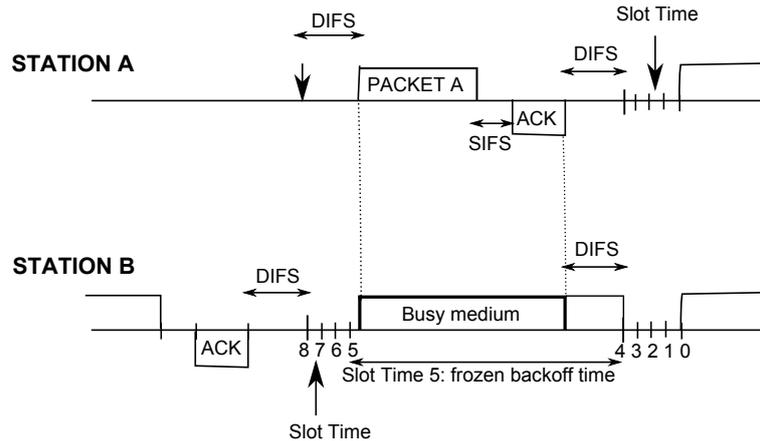


FIGURE 2.4 – Mécanisme d'accès basique du IEEE 802.11 DCF

occupé pour les autres stations jusqu'à la réception de l'ACK. Si l'ACK n'est pas détecté par la station émettrice, une retransmission sera envisagée conformément aux règles du backoff comme le montre la figure 2.3.

2.4.2.2 Modélisation stochastique des réseaux sans fil

Comme le mécanisme du backoff est aléatoire, des méthodes stochastiques ont été développées pour analyser le comportement du réseau sans fil pour définir les paramètres de la QoS. Des études ont modélisé la dynamique de ce type de réseaux et leur comportement stochastique. [Bia00] a étudié les performances du DCF avec un modèle markovien. Il a analysé les paramètres du réseau sous des conditions de saturation c'est-à-dire lorsque les stations ont toujours des paquets à transmettre. L'évaluation analytique de [Bia00] du réseau est faite sous l'hypothèse d'un canal de communication idéal (pas d'interférence, pas de stations cachées). Dans ce cas, une perte de paquet est due à une collision. Le modèle de [Bia00] a abouti à des expressions analytiques du débit de saturation et du retard. Le modèle markovien de la figure 2.5 modélise l'état du paquet dans la procédure du backoff. Soit p la probabilité de collision d'un paquet en transmission, cette probabilité est constante et indépendante de la procédure du backoff et le compteur du backoff est décrémenté à la fin de chaque slot de temps. L'état de transmission d'un paquet peut être décrit par j et k tel que j représente l'étape du backoff avec $j \in [0..R]$ et k correspond au compteur du

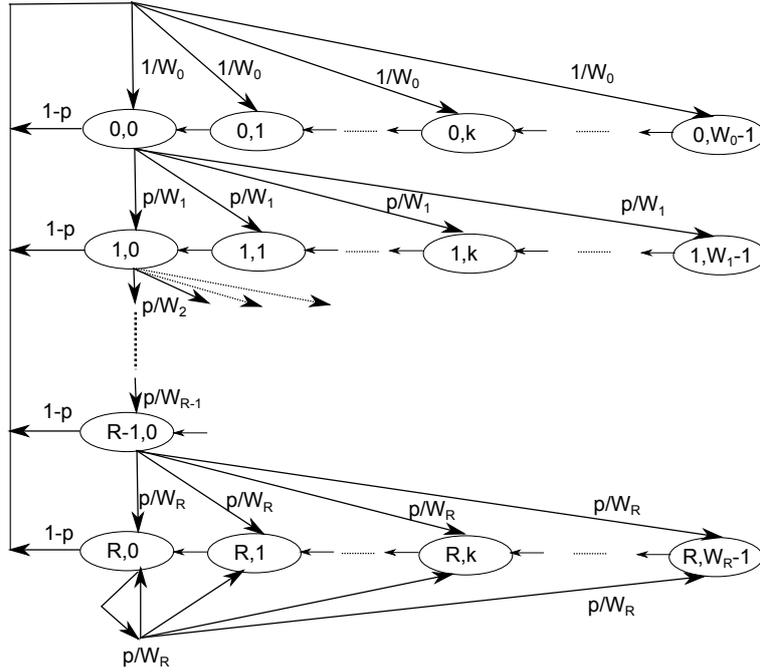


FIGURE 2.5 – Le modèle de Markov de Bianchi [Bia00]

backoff avec $k \in [0..W_j - 1]$ où $W_j = CW_j$ est la taille de la fenêtre de contention dans l'étape j . L'accès aléatoire dans un Wlan est par conséquent représenté par deux processus stochastiques. On considère $b(t)$ le premier processus qui représente le compteur du backoff donnant la durée d'attente aléatoire avant que la station commence la transmission. Soit $s(t)$ le deuxième processus décrivant l'étape actuelle du backoff d'une station à un slot de temps t . $b(t)$ est uniformément distribué dans $[0..W_j - 1]$ de la j^{eme} étape du backoff. Pour décrire toute la procédure du backoff de la figure 2.3, [Bia00] a modélisé le processus bi-dimensionnel par une chaîne de markov à temps discret comme le montre la figure 2.5. La matrice de transition de cette chaîne de markov possède les probabilités de transition suivantes :

$$P\{j, k|j, k + 1\} = 1; k \in [0..W_j - 2]; j \in [0..R] \quad (2.4)$$

$$P\{0, k|j, 0\} = \frac{1-p}{W_0}; k \in [0..W_0 - 1]; j \in [0..R] \quad (2.5)$$

$$P\{j, k|j - 1, 0\} = \frac{p}{W_j}; k \in [0..W_j - 1]; j \in [1..R] \quad (2.6)$$

$$P\{R, k|R, 0\} = \frac{p}{W_R}; k \in [0..W_R - 1] \quad (2.7)$$

L'équation 2.4 indique qu'à chaque début d'un slot de temps, le compteur du backoff est décrément. L'équation 2.5 montre qu'une nouvelle transmission après la transmission réussie d'un paquet commence avec un backoff de valeur 0 et que le temps du backoff est initialement uniformément choisi dans l'intervalle $[0..W_0 - 1]$. L'équation 2.6 décrit le cas d'une

transmission non réussie. Si la transmission a échoué à l'étape $j - 1$ du backoff, la valeur du backoff s'incrémente et le nouveau temps du backoff est choisi dans $[0..W_j]$. Finalement, l'équation 2.7 modélise le fait que dans le cas où le backoff arrive à la valeur maximale R , le backoff ne s'incrémente plus dans les tentatives ultérieures pour la transmission du paquet. On considère $b_{i,k} = \lim_{t \rightarrow +\infty} Pr\{s(t) = j, b(t) = k\}$ la distribution stationnaire de la chaîne de Markov. [Bia00] a établi les relations entre les $b_{i,k}$ en fonction de $b_{0,0}$ et de la probabilité de collision p que nous allons revoir dans le chapitre 4. Il a ensuite défini la probabilité p_τ qu'une station transmette pendant un slot de temps choisi aléatoirement en fonction de $b_{0,0}$ et p en supposant qu'une transmission se produit quand le compteur du backoff atteint la valeur 0 dans chaque étape du backoff. La probabilité p_τ dépend de la probabilité de collision p qui est déterminée en supposant que la probabilité de collision est la probabilité que, pendant un slot de temps, une des stations restantes transmette. On obtient alors une expression analytique de p en fonction de p_τ et du nombre de stations n . [Bia00] a ainsi déterminé analytiquement le débit de saturation S et dans un travail ultérieur le retard moyen en fonction de p_τ et n [TBX10]. L'étude de [Bia00] puis de [TBX10] montre qu'un retard important peut se produire à cause du protocole de communication IEEE 802.11 et du mécanisme d'évitement de collision. Un retard important, qui n'est ni déterministe ni constant, peut rendre le système non stable.

2.4.2.3 Discussion

Plusieurs extensions du modèle de [Bia00] ont été établies et des corrections ont été apportées au modèle pour le rendre plus conforme aux spécifications du DCF et en particulier l'arrêt de la décrémentation du compteur du backoff (backoff freezing). Pour cette raison, [ZA02] ont rajouté une probabilité de décrémentation puis [F+05] ont montré le manque de précision d'un tel modèle. [TBX10] ont montré que tous les modèles utilisant une probabilité de décrémentation ne sont pas aussi performants que le modèle de [Bia00]. Ils ont ensuite amélioré le modèle de base en introduisant le concept de slots de temps anormaux/irréguliers (*anomalous slots*) qui suivent une transmission réussie. Ils proposent que dans un slot de temps anormal, la probabilité d'une transmission ou d'une collision d'un paquet soit différente de celle d'un autre slot. En effet, la station qui émet un paquet transmis avec succès est la seule station qui peut accéder et utiliser le slot qui vient juste après cette transmission réussie. Ils ont montré la précision d'un tel modèle par rapport aux autres et par rapport à celui de [Bia00]. Mais un tel modèle est difficile à manipuler pour résoudre et déterminer les valeurs des probabilités de collision et de transmission et par la suite les expressions du débit de saturation et du retard moyen. Ils ont analysé celui de [Bia00] en montrant que c'est un modèle simple et qu'il est le plus précis parmi tous les modèles de backoff freezing. Par ailleurs, le modèle de [TBX10] n'est valide que pour les conditions de saturation, i.e. lorsqu'une station a toujours un paquet à transmettre ce qui

n'est pas le cas d'une application temps réel périodique. Pour cela, nous proposons dans la suite un modèle markovien qui sera plus fidèle au comportement du trafic d'une boucle de commande.

2.4.3 Influence de la qualité de service sur la qualité de commande

La conception d'un WNCS est une tâche difficile parce que les systèmes de contrôle ont des exigences strictes en terme de temps et de fiabilité (*reliability*). La QoS est le paramètre d'évaluation de la performance et de la stabilité de la boucle de commande implémentée via le réseau. Pour ce type de systèmes, si les paquets n'arrivent pas avant le *deadline*, les performances du système vont se dégrader. Les réseaux sans fil manquent de fiabilité à cause de leur comportement stochastique comme cela a été modélisé dans la chaîne de Markov de Bianchi et également à cause de la mobilité des nœuds communicants. L'intégration du réseau dans la boucle de commande d'un WNCS engendre plusieurs problèmes. En effet, l'état du réseau en terme de QoS affecte l'état de la QoC. Plusieurs paramètres, qui représentent la QoS du réseau et que nous allons détailler par la suite, ont un impact sur la performance du WNCS et sa stabilité. Il est important d'analyser et comprendre l'influence qu'ont les défaillances du réseau sur le système.

2.4.3.1 Influence du retard

L'étude de l'impact du retard sur la performance d'un WNCS est l'objectif de plusieurs travaux de recherche [Ber09]. Dans un WNCS, les deux retards τ_{sc} et τ_{ca} ne sont pas constants, ils sont variables et stochastiques [GTA11][SSA10][TC03] [ZGK13][ZHY16] parce qu'ils dépendent de la dynamique du réseau. Pour les retards indéterministes, la variation du retard n'est pas connu a priori mais en temps réel. Les retards stochastiques sont décrits par des analyses statistiques en définissant l'état du retard actuel en fonction du retard précédent ou bien en décrivant la corrélation entre le retard actuel et les retards précédents [ZGK13]. Un long retard, qui dépasse la période d'échantillonnage du système, provoque l'instabilité et la dégradation de la performance du système. Pour garantir la stabilité d'un WNCS, il faut étudier la marge de stabilité en fonction du retard en définissant des bornes minimum et maximum [SSA10], [HNX07] et utiliser des méthodes de prédiction de contrôle et de compensation du retard [TC03], [GC10], [GC08], [Gon+16]. [LMT02] ont étudié la performance du système en analysant l'intégrale de l'erreur absolue (IAE) et l'intégrale de l'erreur absolue pondérée par le temps (ITAE). Ils ont utilisé la marge de phase et la bande passante du système pour définir le retard maximal acceptable par le système et ils ont montré l'impact de l'augmentation du retard sur la performance du système. [HNX07] ont défini la valeur maximale de l'erreur de l'état acceptable pour définir la valeur correspondante du retard. [GC08] a modélisé l'impact du retard sur un agent mobile dont la

mission est de suivre une trajectoire. Cet agent a totalement dévié de sa trajectoire à cause du retard engendrant une instabilité. [Mec10] a également analysé l'impact du retard, qui provient d'un réseau chargé, sur un robot mobile commandé en réseau pour trois types de missions : atteinte de cible, suivi de trajectoire et évitement d'obstacle. [TC03] ont montré comment un retard introduit dans la boucle de commande d'un régulateur PI peut avoir un impact sur la stabilité du système (en termes de dépassement et de temps de réponse). Le retard peut en effet réduire la marge de stabilité et déstabiliser le système.

2.4.3.2 Influence des pertes de paquets

Les pertes de paquets ont un impact très important sur un WNCS [ZHY16]. A un instant bien défini, si le paquet de contrôle n'est pas disponible auprès des actionneurs, le paquet est considéré comme perdu. Si cette perte atteint un taux très important (à définir), le système n'est plus stable. L'étude de l'impact des pertes de paquets sur les performances du système est intéressante et primordiale surtout dans le cas de l'utilisation d'un réseau sans fil. Quand le réseau est en mauvais état à cause d'une surcharge de trafic, d'une transmission qui n'arrive pas avant le *deadline* ou une erreur de transmission, le paquet est perdu. Les pertes de paquets, comme le retard, sont indéterministes et aléatoires à cause de la dynamique du réseau. [MKT09] ont étudié l'impact des pertes de paquets sur la QoC et ont défini les marges acceptables de pertes de paquets pour garantir la performance du robot mobile dans sa mission. Pour cela, il existe trois intervalles de taux de pertes de paquets selon les états de la QoS : bonne, dégradée, mauvaise. [GTA11] ont analysé l'influence de la perte de paquets sur le dépassement et la réponse du système à une entrée échelon. Ils ont étudié cet impact dans le régime transitoire de la réponse du système et le degré de dégradation pendant cette phase critique de la dynamique du système. [Ber09] a réduit les défaillances du réseau au seul problème des pertes de paquets. Il a étudié l'impact des pertes de paquets sur la boucle de commande d'un moteur à courant continu en analysant les valeurs du dépassement. Cette étude est centrée sur le diagnostic d'un défaut réseau (la non délivrance de données pendant un temps préfixé) quand on atteint 50% de pertes dans le réseau. [HNX07] ont proposé une méthode stochastique pour modéliser le processus de pertes de paquets en lui attribuant une loi de Bernoulli. Ils ont ensuite déterminé la limite des pertes de paquets garantissant la stabilité du système. En supposant avoir une distribution de Bernoulli, cela implique implicitement que la prédiction de taux de pertes de paquets est certaine ce qui ne correspond pas en pratique à la dynamique des réseaux sans fil. Cette étude a des limites et peut ne pas garantir la stabilité et la performance d'un SCR [Lu+12]. Il est ainsi difficile d'avoir une loi qui décrit le processus de pertes de paquets. Pour cela, des méthodes de contrôle robuste pour un système incertain ont été adoptées [Qu+16].

2.4.3.3 Influence de la gigue

Dans les SCRs, qui sont des systèmes temps réel, la gigue existe toujours puisque les composants du SCR communiquent via le réseau. On peut définir une tolérance sur l'hypothèse qui décrit la contrainte temps réel dans l'équation 2.1. Cette tolérance n'est que la gigue qui définit l'intervalle acceptable de variation du retard par rapport au *deadline*. Les caractéristiques de la gigue sont définies en étudiant conjointement les performances du réseau et la performance de la boucle de commande. La gigue peut être causée par la surcharge du trafic réseau ou bien par la politique d'ordonnancement local des tâches. [SSA10] ont proposé une définition de la gigue : c'est est une désynchronisation par rapport aux instants d'échantillonnage, en fonction de l'erreur de l'amplitude dans le régime transitoire, de la période d'échantillonnage et du rapport signal sur bruit.

2.4.3.4 Influence de la connectivité

Un WNCS mobile présente plus de contraintes qu'un WNCS non mobile. En effet, la qualité du signal en terme de *Received Signal Strength Indication* (RSSI) a une influence sur la transmission correcte des données. Un système mobile peut perdre la connexion avec la station s'il dépasse une limite de distance avec la station de contrôle (la limite de la zone de couverture) ou s'il y a des obstacles qui peuvent provoquer la coupure de la communication [Mec10]. La mobilité et la présence des obstacles peuvent causer l'évanouissement du signal (signal fading). La zone de propagation du signal est composée de trois types de zones comme le montre la figure 2.6 :

- **une zone de transmission** : dans cette zone, le taux d'erreur est faible et la communication est possible sous condition que le RSSI (en dBm) soit supérieur à un seuil bien défini. Ce seuil garantit la transmission correcte du signal des données.
- **une zone de détection** : le signal peut être détecté dans cette zone mais la communication n'est pas possible.
- **une zone d'interférence** : le signal ne peut pas être détecté dans cette zone et il est considéré comme un bruit.

2.5 Conclusion

L'utilisation des réseaux sans fil, bien qu'ayant des avantages, apporte des contraintes spécifiques. L'intégration, dans la boucle de commande, du réseau sans fil engendre l'indéterminisme dans la transmission des données. Les paramètres de la QoS (retard, pertes de paquets, gigue, connectivité, etc) ont un impact non négligeable et il est parfois difficile à les modéliser analytiquement à cause de la dynamique des réseaux sans fil. Dans les travaux

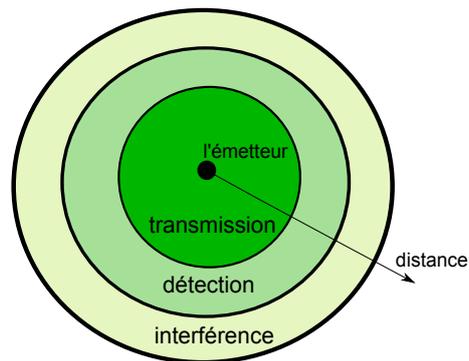


FIGURE 2.6 – Les zones de propagation du signal [Dho03]

de l'automatique, l'étude de la stabilité des systèmes est analytique mais la dégradation stochastique de la QoS et son impact sur la QoC peuvent ne pas être représentables par un modèle mathématique précis. Dans ce cas, le recours aux méthodes probabilistes est une solution pour garantir la performance des WNCS. Le chapitre 5 va présenter une méthode probabiliste qui permettra de modéliser la causalité entre la QoS et la QoC.

Réseaux bayésiens distribués et diagnostic

Sommaire

3.1	Introduction	27
3.2	Réseaux bayésiens	28
3.2.1	Définition	28
3.2.2	Indépendance conditionnelle et d-séparation	29
3.3	Raisonnement avec les réseaux bayésiens	30
3.4	Inférence dans les réseaux bayésiens	31
3.4.1	Types d'inférences	31
3.4.2	Types d'observations/évidences	32
3.5	Raisonnement distribué dans les réseaux bayésiens	34
3.5.1	Réseaux bayésiens distribués : état de l'art	34
3.5.2	Réseaux bayésiens modulaires	43
3.6	Réseaux bayésiens pour le diagnostic	47
3.7	Réseaux bayésiens pour le diagnostic distribué	51
3.8	Conclusion	53

3.1 Introduction

Pour les systèmes à comportement stochastique, les études probabilistes sont pertinentes pour décrire les événements probabilistes qui représentent leurs dynamiques. Étudier les incertitudes et quantifier la fréquence d'occurrence d'événements sont les objectifs des études probabilistes. De nos jours, les modèles probabilistes sont utilisés pour le diagnostic, le pronostic, la sûreté de fonctionnement et la maintenance. Ces modèles servent à l'aide à la décision, à la surveillance de l'état du système, à l'étude de l'impact des perturbations et de l'environnement sur le système, à la prévision des pannes avant leurs occurrences et

à l'étude de l'impact de la maintenance sur le fonctionnement du système [WP17]. Les méthodes probabilistes comme les RBs, les chaînes de Markov, ou les diagrammes d'influence, sont des outils de l'intelligence artificielle qui sont fréquemment utilisés en industrie et qui ont montré leur efficacité dans plusieurs domaines d'applications pour la détection de défaut et l'identification ainsi que l'analyse de la sûreté de fonctionnement. Les RBs en particulier, qui représentent des dépendances causales entre des événements, représentent un outil de modélisation et d'analyse performant [WP17]. En outre, ils assurent une analyse a posteriori des états à partir d'une collection d'évidences. Un RB est un outil adéquat pour notre problématique de diagnostic d'un SCR sans fil dont le comportement est stochastique à cause de la dynamique des réseaux sans fil. Dans ce chapitre, nous présentons les RBs et toutes leurs caractéristiques qui vont servir à atteindre notre objectif d'études. Les RBs distribués et modulaires sont aussi présentés dans ce chapitre en détaillant leur utilité pour la modélisation de systèmes modulaires distribués tels que les SCRs sans fil.

3.2 Réseaux bayésiens

3.2.1 Définition

Les réseaux bayésiens RBs sont des modèles graphiques probabilistes mis en œuvre dans le diagnostic utilisant l'intelligence artificielle. Ils représentent les dépendances entre événements ainsi que leurs probabilités d'apparition. Le principal avantage que possède ce modèle est la possibilité de rassembler et de fusionner des données de diverses natures dans un même modèle [Naï+07] : retour d'expérience, expertise (règles logiques, équations, statistiques ou probabilités) et observations. Les réseaux bayésiens, dont le nom provient des travaux de Thomas Bayes sur la théorie des probabilités, sont le résultat de recherches initiées dans les années 1980 par [Pea88]. Un réseau bayésien est défini par [Naï+07] :

- un graphe acyclique orienté G , $G = (V, E)$ où V est l'ensemble des nœuds et E est l'ensemble des arcs
- un espace probabilisé fini (Ω, Z, P)
- et un ensemble de variables aléatoires associées aux nœuds du graphe tel que :

$$P(V_1, V_2, \dots, V_n) = \prod_{i=1}^n P(V_i | C(V_i))$$
où $C(V_i)$ sont les parents de V_i

Pour construire un RB, il faut suivre trois étapes :

1. **étape qualitative** : c'est une étape qui consiste à définir les variables et leur espace d'état. Ces variables, discrètes ou continues, sont définies à l'aide des experts et leur espace d'états correspond à leurs valeurs possibles. Il faut aussi définir la structure du RB après avoir validé l'ensemble des variables : cela consiste à relier les variables par des arcs orientés qui indiquent la relation de cause-à-effet entre elles.
2. **étape quantitative** : cette étape consiste à définir les tables de probabilités de

chaque variable X_i qui vont, ensuite, caractériser la loi de probabilité jointe du RB. Pour fixer ces lois de probabilités, les avis des experts sont collectés si les données sur les variables ne sont pas disponibles. Dans le cas où les données sont disponibles, un apprentissage automatique des probabilités du RB est effectué à l'aide de l'algorithme du maximum de vraisemblance. C'est un algorithme qui est basé sur l'estimation statistique des probabilités d'un évènement à partir de sa fréquence d'occurrence dans l'ensemble de données utilisé dans l'apprentissage [FGG97]. Cette estimation de probabilité est définie par l'équation 3.1 :

$$P(X_i = x_k | Parents(X_i) = x_j) = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}} \quad (3.1)$$

avec $N_{i,j,k}$ le nombre d'occurrences où $X_i = x_k$ et $Parents(X_i) = x_j$

3.2.2 Indépendance conditionnelle et d-séparation

La d-séparation dans un RB est une règle qui décrit les conditions dans lesquelles les informations entre un nœud X et un nœud Y sont bloquées. C'est une propriété graphique qui agit sur la sémantique de causalité d'un graphe. X et Y sont par définition d-séparés par Z , si pour tous les chemins entre X et Y , l'une au moins des deux conditions suivantes est vérifiée [Naï+07] :

- le chemin converge en un nœud W , tel que $W \neq Z$, et W n'est pas une cause directe de Z .
- le chemin passe par Z , et est soit divergent, soit en série au nœud Z .

Si X et Y sont d-séparés par Z cela signifie que Z bloque le passage de l'information entre X et Y dans le cas où Z est la seule information connue dans le graphe.

- **$\mathbf{X} - > \mathbf{Z} - > \mathbf{Y}$ (structure en série)** : l'information ne peut circuler de X à Y que si Z est inconnu. Si Z est la seule information connue, une connaissance sur X ne modifiera en rien la connaissance sur Y . Donc, le circuit d'information de X à Y est bloqué par Z .
- **$\mathbf{X} < -\mathbf{Z} - > \mathbf{Y}$ (structure divergente)** : l'information ne peut circuler de X à Y que si Z n'est pas connu. Si on a une connaissance sur Z , on aura une information sur l'état de Y et l'information qu'on peut avoir sur X n'y change rien.
- **$\mathbf{X} - > \mathbf{Z} < -\mathbf{Y}$ (structure convergente)** : l'information ne peut circuler de X à Y que si Z est connue. Une connaissance sur X n'a aucun lien à priori avec l'état de Y . Mais si on a de plus une connaissance sur Z , la connaissance sur Y va changer. Donc, un chemin de circulation d'information de X à Y est ouvert.

Les trois types de structure (en série, convergente et divergente) sont présentés sur la figure 3.1.

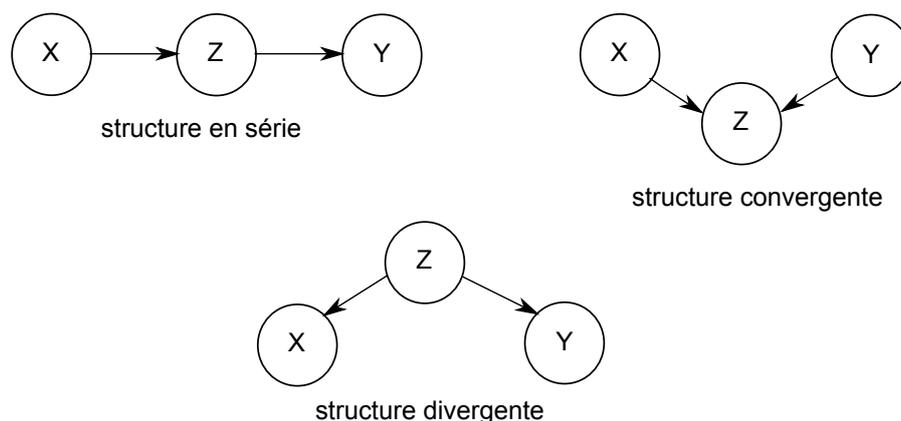


FIGURE 3.1 – Types de structures d'un RB

3.3 Raisonnement avec les réseaux bayésiens

Le RB représente des dépendances de cause-à-effet entre des variables auxquelles on associe initialement une probabilité a priori. Si de nouvelles informations/observations apparaissent sur certaines variables (les variables d'observations ou d'évidences), l'algorithme d'inférence calcule les probabilités a posteriori des autres variables non observées en mettant à jour leurs probabilités marginales. Une requête permet d'obtenir la probabilité marginale d'une variable du RB (voir annexe A).

Il existe trois types de raisonnement dans un RB [KN03] : le diagnostic, le pronostic et le raisonnement combiné. Pour un raisonnement de diagnostic, les nœuds de type symptômes/effets reçoivent des évidences et à l'aide de l'algorithme d'inférence, les probabilités a posteriori des causes sont mises à jour. Une requête appliquée sur les nœuds de type causes donnent le résultat du diagnostic comme le montre la figure 3.2. Ce raisonnement se produit dans le sens inverse des arcs dirigés du RB. Un autre type de raisonnement est celui de prédiction/pronostic, qui consiste à appliquer des évidences/observations sur les causes et à calculer les nouvelles probabilités a posteriori des effets : c'est un raisonnement dans le sens des arcs dirigés comme l'indique la figure 3.2. Un raisonnement combiné peut se produire dans le RB comme le montre la figure 3.2 : combinaison du diagnostic et pronostic. Un tel raisonnement peut être adapté pour un raisonnement de diagnostic comme cela est détaillé dans le modèle bayésien pour le diagnostic de [PC07].

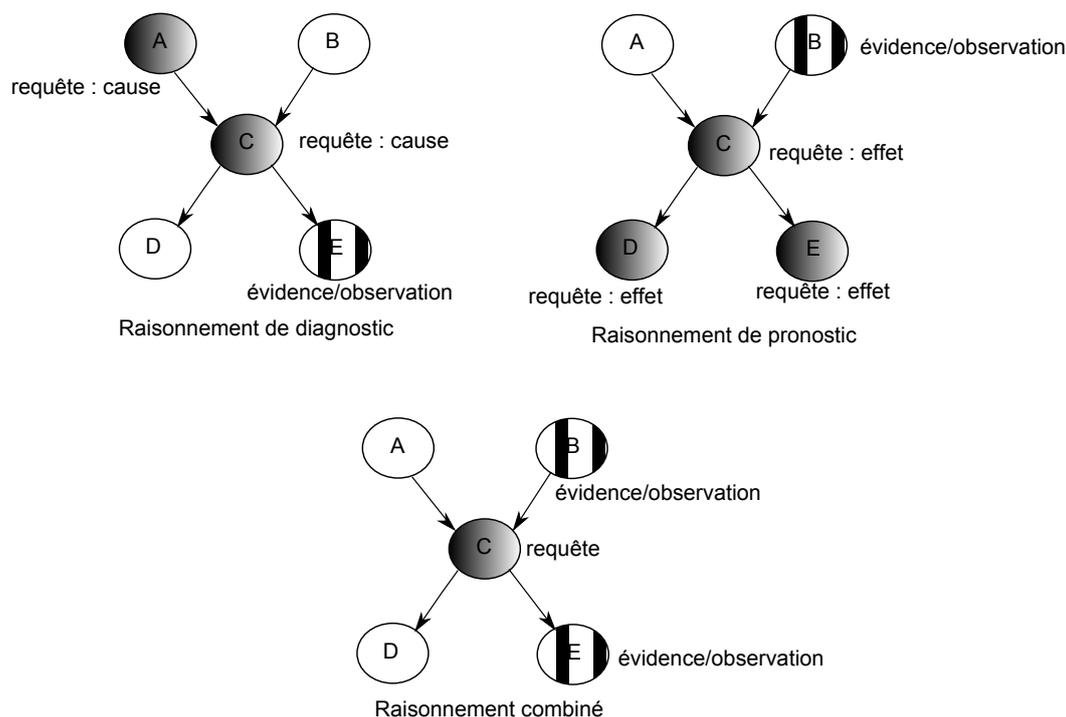


FIGURE 3.2 – Types de raisonnement

3.4 Inférence dans les réseaux bayésiens

3.4.1 Types d'inférences

L'inférence dans les RBs consiste à calculer les probabilités a posteriori des variables à partir d'un ensemble d'évidences sur des variables observables, c'est une procédure qui sert à mettre à jour les croyances. Dans un RB constitué d'un ensemble de nœuds $\{X_1, X_2, \dots, X_n\}$, l'algorithme d'inférence calcule les probabilités a posteriori $P(X_i|Y)$ de la variable non observable X_i sachant les observations Y avec $X_i \neq Y$. Il existe deux types d'inférence : l'inférence exacte, que nous avons détaillée dans l'annexe A, et l'inférence approximative. Pour l'inférence exacte, [Pea88] a développé un algorithme de passage de messages entre les variables du RB. En effet, un nœud ou une variable X_i reçoit les messages λ qui proviennent des nœuds fils et les messages π qui sont produits par ses parents. Chaque variable est un producteur de ces deux types de messages. Cet algorithme consiste à mettre à jour la distribution de probabilité de proche en proche pour chaque nœud du RB à l'aide des messages provenant de ses voisins pour transmettre les observations. Il ne peut être appliqué que sur des RBs dont le graphe est un arbre ou une forêt d'arbres. Un arbre est défini comme étant un graphe :

- **connexe** : car il existe un chemin (succession de liens) entre tout couple de nœuds ;

- **sans cycles** : car il est impossible de partir d'un nœud et d'y revenir sans franchir deux fois le même lien.

Pour les RBs dont les graphes ne sont pas des arbres, *Finn V. Jensen* a établi l'algorithme de l'arbre de jonction [HD96]. Cet algorithme modifie la structure du RB pour la transformer en une structure en arbre puis applique la procédure de passage de messages de [Pea88]. Il existe trois étapes pour aboutir à une structure en arbre :

1. **la moralisation** : c'est une étape qui consiste à construire un graphe moral qui est un graphe non orienté. La moralisation consiste à enlever les directions des arcs pour obtenir des arêtes et relier les parents de chaque nœud pour obtenir des cliques. Une clique est un ensemble de nœuds tel qu'il existe un lien reliant tout couple de variables de cet ensemble.
2. **la triangulation** : cette étape consiste à connecter les nœuds non adjacents dans les cycles du graphe moral de plus de trois nœuds pour obtenir un graphe triangulé.
3. **l'arbre de jonction** : cette étape consiste à construire l'arbre de cliques, c'est-à-dire que chaque nœud de l'arbre représente une clique du graphe triangulé. Les liens entre les nœuds de l'arbre de jonction vérifient la propriété d'intersection courante. La propriété d'intersection courante signifie que dans toute chaîne reliant deux cliques C_1 et C_2 de l'ensemble de cliques, toutes les cliques de la chaîne contiennent $C_1 \cap C_2$ [NG04]. Cette propriété permet de définir l'ensemble des nœuds de séparation.

Les nouveaux paramètres attribués aux cliques du nouveau graphe, c'est-à-dire la distribution de probabilité de chaque clique, sont définis comme suit : pour chaque nœud X du graphe initial, choisir une clique E qui contient tous les parents de X , la nouvelle valeur du potentiel ou de la probabilité ϕ_E de E est égale à $\phi_E = \phi_{E,\text{précédent}} \times P(X|\text{parents}(X))$, avec $\phi_{E,\text{précédent}}$ la valeur du potentiel de E avant sa mise à jour. Cette méthode est la meilleure solution connue en terme de complexité algorithmique. Il a été démontré que le problème d'inférence dans un RB est NP complet. Les algorithmes d'inférence approximative peuvent résoudre les problèmes d'inférence dans les RBs de grande taille [Naï+07].

3.4.2 Types d'observations/évidences

L'inférence rend les RBs très intéressants pour calculer les probabilités a posteriori sachant un ensemble d'observations. Les observations sont les états des variables observables à un instant donné. Il existe deux types d'évidences ou observations [KN03] : l'observation certaine et l'observation incertaine.

L'observation certaine est une observation déterministe e sur une variable X , telle que la valeur de la probabilité d'avoir $X = x$ est 1 et les probabilités des autres états de X sont nulles. [Pea88]. Cette observation est une instantiation de X avec la valeur x telle que $P(X = x|e) = 1$. Dans le cas général, les observations sont tout le temps certaines. Les tables de probabilités conditionnelles et les probabilités marginales (les croyances sur les

variables) représentent les incertitudes dans les RBs. Quand une nouvelle observation e est appliquée, les croyances sur les variables non observées vont être mises à jour en calculant les probabilités a posteriori. La nouvelle croyance $P(Y|e)$ sur une variable Y est la nouvelle valeur de l'incertitude sur l'état de Y .

3.4.2.1 Observation incertaine

Il existe deux causes majeures qui engendrent l'incertitude dans les observations. L'observation peut subir une imprécision à cause d'erreurs et de bruits de mesures ou à cause d'une vision partielle ou locale sur la variable d'observation concernée [Mra+15]. Dans ce cas, l'observation ne peut plus être déterministe, elle est considérée comme incertaine. Parmi les grandes familles d'observations incertaines, on trouve les observations de vraisemblance ou virtuelle ("*likelihood evidence, virtual evidence*") introduits pour la première fois dans [Pea88]. La question qui se pose est la suivante, comment les algorithmes d'inférence vont traiter les observations incertaines ?

[Pea88] a défini une méthode d'inférence pour propager une observation incertaine sur une variable X en utilisant un nœud virtuel V . Ce nœud est un nœud auxiliaire qui sert à transformer une observation incertaine sur X en une observation certaine sur V , puis à utiliser les algorithmes d'inférence classiques (arbre de jonction). Le nœud V est ajouté comme nœud fils de X . Il possède deux états possibles : *oui*, *non*. Prenons l'exemple simple de la figure 3.3 pour illustrer cette notion d'évidence virtuelle. Soient un nœud X dont la distribution de probabilité est $\{vrai, faux\} = \{0.5, 0.5\}$ et V son nœud fils virtuel (voir figure 3.3). L'incertitude dans l'observation de X est de 80%. Elle est représentée dans la table de probabilités conditionnelles de V avec $P(V = oui|X = vrai) = 0.8$ et $P(V = oui|X = faux) = 0.2$. On applique une évidence certaine telle que $V = oui$. Les probabilités a posteriori de X sont calculées dans les équations suivantes en utilisant le théorème de Bayes :

$$\begin{aligned}
 P(X = vrai|V = oui) &= \alpha P(V = oui|X = vrai)P(X = vrai) \\
 &= \alpha 0.8 \times 0.5 \\
 P(X = faux|V = oui) &= \alpha P(V = oui|X = faux)P(X = faux) \\
 &= \alpha 0.2 \times 0.5
 \end{aligned} \tag{3.2}$$

Sachant que $P(X = vrai|V = oui) + P(X = faux|V = oui) = 1$, le paramètre α est égal à 2 et par la suite $P(X = vrai|V = oui) = 0.8$ et $P(X = faux|V = oui) = 0.2$.

En pratique, pour présenter l'observation incertaine, on n'utilise pas l'observation de vraisemblance, c'est le rapport de vraisemblance qui est utilisé : $\frac{P(V=oui|X=vrai)}{P(V=oui|X=faux)}$. Cela revient à dire que l'état $X = vrai$ est quatre fois plus probable que $X = faux$. Ce rapport doit être bien défini pour avoir des probabilités a posteriori adéquates c'est-à-dire des valeurs de $P(X = vrai|V = oui)$ et $P(X = faux|V = oui)$ égales aux incertitudes observées.

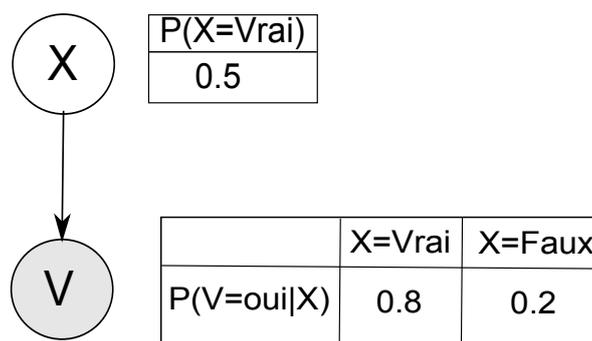


FIGURE 3.3 – Evidence virtuelle [KN03]

3.5 Raisonnement distribué dans les réseaux bayésiens

Les RBs permettent de modéliser des relations de dépendances entre des données diverses et de domaines hétérogènes. Dans des environnements dynamiques et incertains, de nouvelles sources d'informations (ou de nouveaux domaines) peuvent intégrer le système global et interagir avec lui. Les sources d'informations sont généralement distribuées quand le système est réparti physiquement. Des informations hétérogènes et distribuées ont ainsi besoin d'être traitées rapidement. Dans ce cas, il faut utiliser les ressources disponibles tel que le réseau de communication pour transmettre toutes les informations à une unité centrale. Il existe alors un risque de surcharger le réseau de communication. En outre, un large RB qui décrit des dépendances entre plusieurs domaines complexes nécessite l'avis de différents experts. C'est ainsi que le RB distribué s'avère comme étant une bonne solution pour des systèmes complexes, distribués, mobiles et dynamiques. En effet, les RBs distribués permettent une analyse modulaires par domaines d'expertises.

3.5.1 Réseaux bayésiens distribués : état de l'art

Les RBs distribués ont été utilisés dans plusieurs travaux dans le but de réduire le temps d'inférence, la charge sur le serveur de calcul et d'avoir la possibilité d'inférer des informations de différents environnements. En effet, les informations provenant de dispositifs distants doivent être pré-traitées localement dans des modules bayésiens [Fra+14]. Un module bayésien est un sous-graphe ou un sous-RB du RB global. Les résultats d'inférences locales de différents modules bayésiens sont généralement rassemblés pour une dernière inférence. Il existe trois catégories de réseaux bayésiens distribués [Fra+14] qui ont été étudiés pour choisir l'approche la plus adaptée à notre système et à notre contexte de travail :

1. **catégorie 1** : les modules bayésiens ont une structure commune pour l'inférence.

2. **catégorie 2** : les modules bayésiens doivent former un graphe acyclique et ils sont structurés en une forêt de jonction (forêt d'arbres de jonction). Une moralisation et une triangulation distribuées sont appliquées sur tous les agents bayésiens pour assurer la consistance entre les modules bayésiens et la construction de l'arbre de liaison entre les arbres de jonction locaux.
3. **catégorie 3** : les modules RBs sont indépendants. L'inférence est complètement indépendante et distribuée.

Dans la **première catégorie**, une seule structure commune d'inférence est utilisée où l'inférence distribuée n'est pas possible comme dans le cas des modèles bayésiens de [LM97], [PG04] et [PGM05]. Toutes les informations doivent être traitées sur une unité centrale, une situation qui ne permet pas d'optimiser l'utilisation des ressources de communication [Fra+14]. Les travaux de [PG04] et [PGM05] ont proposé une méthode d'inférence pour les systèmes distribués et plus précisément d'inférence dans un réseau de capteurs de températures. Un RB est associé à chaque capteur comme le montre la figure 3.4. Les RBs doivent

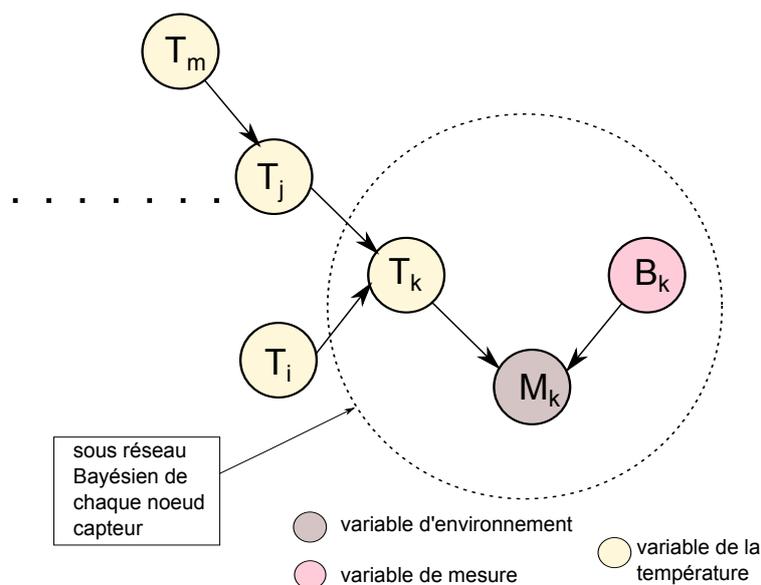


FIGURE 3.4 – Réseau Bayésien d'un réseau de capteurs [PGM05]

ensuite être organisés sous la forme d'un arbre de jonction pour l'inférence. Chaque RB, correspondant à un capteur, contient trois types de variables : une variables de mesure M_i observable, une variable non observable T_i qui indique la vraie valeur de la température et une variable non observable d'environnement B_i qui caractérise l'environnement dans lequel le capteur fonctionne. Une corrélation existe entre les températures données par chaque nœud (capteur) pour définir les liens entre les différents RBs associés aux nœuds du réseau de capteurs. Sur la figure 3.4, il existe ainsi une corrélation entre T_j et T_k d'une part et T_i et T_k d'autre part. La densité de probabilité jointe dans tout le réseau des RBs

est alors définie comme suit :

$$P(T_1, \dots, T_n, B_1, \dots, B_n, M_1, \dots, M_n) = \prod_{i,j} P(T_i, T_j) \prod_{k=1}^n P(B_k)P(M_k|B_k, T_k) \quad (3.3)$$

avec $P(T_i, T_j)$ la probabilité jointe a priori des nœuds T_i et T_j voisins.

L'inférence distribuée est définie de la façon suivante : après avoir obtenu les observations associées à ses variables de mesures, un RB, qui correspond à un nœud capteur, doit collaborer avec le reste du réseau pour obtenir la probabilité a posteriori de son nœud T_i . Cette probabilité a posteriori (requête) $P(T_i|m_1, \dots, m_n)$ est obtenue compte tenu des mesures m_i faites par tous les nœuds du réseau, où n représente le nombre de capteurs dans le réseau. Pour effectuer l'inférence, il faut construire l'arbre de jonction qui va permettre d'avoir une convergence des résultats d'observations.

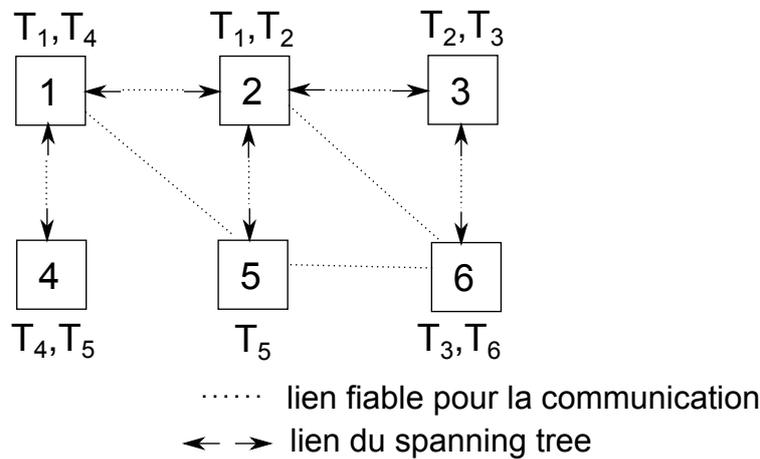


FIGURE 3.5 – Exemple d'un spanning tree [PGM05]

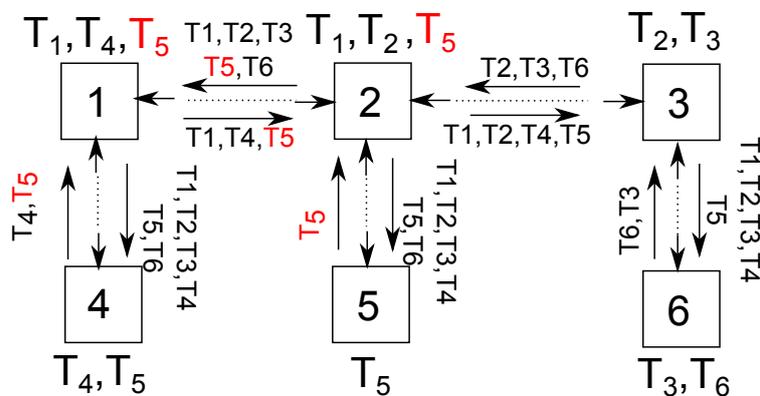


FIGURE 3.6 – Échange de messages dans un spanning tree [PGM05]

Pour obtenir l'arbre de jonction, [PGM05] ont établi une stratégie qui consiste en deux parties :

1. **construction d'un *spanning tree*** : chaque nœud capteur désigne un ensemble de nœuds capteurs voisins avec qui il a un lien de communication en bon état. Les nœuds capteurs sont organisés sous forme de *spanning tree* pour éviter d'avoir des cycles dans la procédure d'inférence. Les nœuds collaborent pour définir leurs nœuds parents et le nœud racine. Le nœud qui a l'identifiant le plus petit est élu le nœud racine. Par la suite, chaque nœud choisit un parent avec qui il a un lien de haute qualité et qui a un chemin qui mène à la racine. Quand les nœuds se mettent d'accord sur le nœud racine et sur leurs nœuds parents, ils forment alors un *spanning tree*.
2. **construction de l'arbre de jonction** : l'arbre de jonction est la structure de données utilisée dans l'algorithme de propagation de messages pour l'inférence. Pour construire l'arbre de jonction, qui est un arbre de cliques, il faut définir les cliques et les séparateurs des RBs associés aux nœuds capteurs du réseau. Chaque nœud effectue un apprentissage de ses cliques et de ses séparateurs en échangeant des messages avec ses nœuds voisins dans le *spanning tree* comme le montre la figure 3.5 de [PGM05]. Comment obtenons-nous ces messages ? Ces messages vont permettre de définir les cliques vérifiant la propriété de l'intersection courante. Soit un nœud i et son nœud voisin j . Un domaine D_i est associé à chaque nœud i , définissant les variables T_i corrélées pour deux nœuds qui sont liés. Dans l'exemple de la figure 3.5, le domaine du nœud 2 est $\{T_1, T_2\}$ auquel est associée une probabilité a priori $P(T_1, T_2)$. Les messages échangés entre deux nœuds voisins i et j définissent les états des variables que chaque nœud peut atteindre. Les variables que j peut atteindre à partir de i sont définies comme suit :

$$R_{ij} = D_i \cup_{k \in n(i), k \neq j} R_{ki} \quad (3.4)$$

avec $n(i)$ l'ensemble de voisins de i dans le *spanning tree*.

Le nœud i calcule R_{ij} en définissant les variables qui peuvent être atteintes à partir de tous ses voisins sauf j et en ajoutant les variables de son domaine D_i . Il envoie ensuite le message R_{ij} au nœud j . La figure 3.6 montre les messages échangés entre les nœuds du *spanning tree* de la figure 3.5. Dans la figure 3.6, où il y a six nœuds capteurs, le nœud 3 envoie le message $R_{32} = \{T_2, T_3, T_6\}$ au nœud 2, avec $R_{32} = R_{63} \cup D_3$ où $D_3 = \{T_2, T_3\}$ et le nœud 6 est le nœud voisin du nœud 3. En outre, si un nœud reçoit la même variable dans deux messages, le nœud doit inclure cette variable dans sa clique pour vérifier la propriété de l'intersection courante. Et par conséquent, le nœud détermine sa clique de la façon suivante :

$$C_i = D_i \cup_{j, k \in n(i), j \neq k} R_{ji} \cap R_{ki} \quad (3.5)$$

Dans la figure 3.6, le nœud 1 reçoit deux messages qui contiennent T_5 , donc il doit

inclure T_5 dans sa clique. Les séparateurs entre deux nœuds voisins i et j sont alors définis par $S_{ij} = C_i \cap R_{ji}$.

Une inférence exacte est ensuite effectuée dans l'arbre de jonction construit pour le réseau de capteurs. Le problème de cette méthode est qu'elle n'est valable que si le réseau est petit. En effet, la complexité de l'algorithme de l'arbre de jonction augmente de façon exponentielle avec le nombre de nœuds. En plus, nous avons vu que les nœuds doivent être organisés dans une structure bien définie pour assurer la convergence de la procédure d'inférence.

Le modèle de [Xia00] "*Multiply Sectioned Bayesian Network*" (MSBN) est un exemple des RBs distribués qui représentent la **deuxième catégorie**. Le MSBN est un modèle de RBs distribués qui consiste à diviser un grand RB en plusieurs sous réseaux permettant un raisonnement probabiliste entre différents agents. Ce type de formalisme est utilisé pour modéliser un sous domaine à l'aide de sous RBs qui lui sont associés [XL00]. La règle de d -séparation est utilisée pour diviser le RB global. Les sous réseaux résultants, qui sont des graphes acycliques directs DAG, sont organisés sous forme d'un hyper-arbre (*hypertree*). Chaque nœud dans l'hyper-arbre représente un agent et son domaine de connaissance. En d'autres termes, chaque agent contient une partie d'un RB global plus grand et l'union des DAGs de tous les sous-domaines est un DAG. Un agent communique avec ses voisins dans l'hyper-arbre via des messages qui sont les distributions de probabilités jointes des variables qu'ils partagent. Pour les variables partagées, une distribution de probabilité jointe demande la collaboration d'un agent avec les autres. Les interfaces des agents d -séparent les sous RBs [XL00]. Il n'existe qu'une seule distribution de probabilités pour toutes les variables aléatoires des RBs locaux. Ces dernières contraintes limitent l'autonomie des agents qui doivent être strictement couplés. L'inférence distribuée dans un MSBN consiste à transformer l'hyper-arbre MSDAG en une forêt d'arbres de jonction liés, c'est-à-dire un arbre d'arbres de jonction. Une procédure de transmission de messages dans un arbre de jonctions est utilisée pour distribuer les observations et les évidences entre les agents en assurant une cohérence globale dans le système. Intégrer un agent dans la forêt des arbres de jonction s'avère être une autre contrainte pour le MSBN. Pour cela, l'hyper-arbre MSDAG commence par une moralisation et une triangulation distribuée pour une cohérence globale entre les agents. En effet, l'étape de triangulation sert à la construction des arbres de jonction comme le montre la figure 3.7 [Xia00] qui est une structure utilisée pour la communication et le passage de messages. Chaque agent construit un arbre de jonction local par moralisation et triangulation pour une inférence au niveau de son sous-domaine si et seulement si des observations locales sont introduites. Le MSDAG est transformé finalement en forêt d'arbres de jonction liés sur laquelle l'inférence va être effectuée. Pour l'inférence dans un arbre de jonction d'un seul réseau bayésien, le message échangé entre deux cliques C et $C1$ est une table de croyance sur leur intersection, l'ensemble de nœuds séparateurs, appelée le *sepset*. Dans un MSBN, l'intersection de deux sous-domaines, appelé un *d-sepset*, est un arbre de jonction de liaison qui permet le passage de messages

entre deux arbres de jonction adjacents [Xia00].

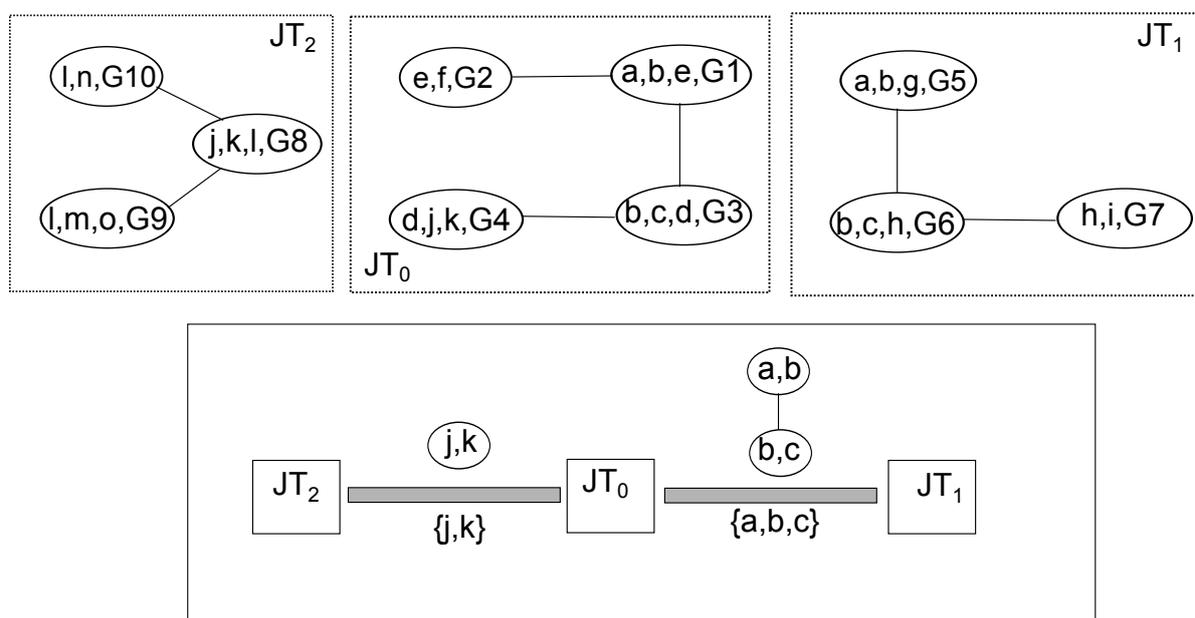


FIGURE 3.7 – Arbre de jonction

Pour une variable partagée entre plusieurs sous-réseaux bayésiens, un seul sous-réseau, celui qui contient toute la famille de cette variable (les variables parents), définit sa probabilité conditionnelle. La méthode de [Xia00] consiste à attribuer la table de probabilités conditionnelles de cette variable à un seul agent et une distribution uniforme aux autres. Si cette condition n'est pas respectée, la décomposition en agents n'est pas respectée et dans ce cas des agents peuvent être fusionnés d'où la modification des caractéristiques du sous-domaine. Le MSBN est une solution pour modéliser les systèmes distribués. En revanche, cette structure n'est pas adaptée aux réseaux sans fil dont la dynamique évolue au cours du temps et les entités communicantes sont mobiles. Le réseau sans fil peut provoquer des retards et des pertes de données aléatoires. Des problèmes sont alors introduits dans la procédure de triangulation et de moralisation distribuées, des désynchronisations et des retards montrent la non robustesse de cette structure.

Les modèles à base d'agents équipés de réseaux [BV02], [LVB10] sont des modèles où chaque agent implémente sa propre connaissance sous la forme d'un RB local. Ils appartiennent à la *deuxième catégorie* des RBs distribués. Chaque RB local possède sa propre distribution de probabilité utilisée par son agent pour son raisonnement interne à lui. Cette structure bayésienne est appelée "*Agent Encapsulated Bayesian Network*" (AEBN). Les agents sont organisés sous une hiérarchie de fournisseur-demandeur d'informations (*publisher-subscriber*) pour un ensemble de variables en fonction des objectifs. La structure ou le graphe de communication des agents doivent être sous la forme d'un graphe

acyclique dirigé DAG. Ils communiquent par des messages via les variables partagées. Le fournisseur est l'expert donc celui qui a la meilleure connaissance de la variable et il partage l'information avec les demandeurs. Les agents demandeurs d'informations mettent à jour leurs croyances en se basant sur leurs propres observations et sur les croyances du fournisseur parce que l'agent fournisseur possède la meilleure connaissance. Et dans ce cas, l'agent demandeur remplace sa vision actuelle sur la variable partagée par celle du fournisseur. Un agent peut être d'une part fournisseur d'informations pour les variables sur lesquelles il a le plus d'expertise et d'autre part un demandeur d'informations pour les variables sur lesquelles il a besoin d'expertise. Un agent équipé d'un RB local est un agent dont la base de connaissances interne est représentée par un RB ayant trois types de variables [LVB10] :

- **variables d'entrées I** : variables pour lesquelles d'autres agents ont une meilleure connaissance.
- **variables de sortie O** : variables pour lesquelles l'agent a les meilleures connaissances et qui sont partagées avec les autres agents.
- **variables locales ou cachées L** : variables privées à l'agent.

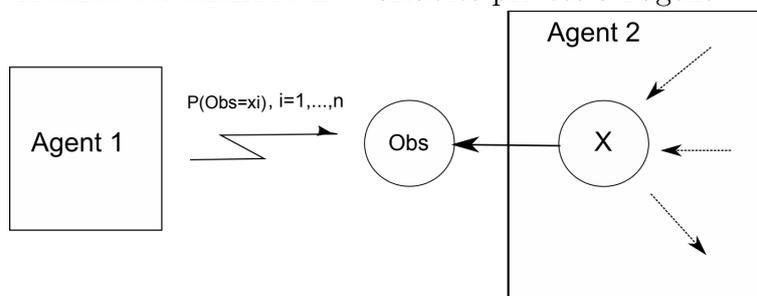


FIGURE 3.8 – Transmission de messages via la variable d'observation dans un AEBN

Les variables $I \cup O$ sont donc des variables partagées entre l'agent et d'autres agents alors que les variables L sont privées et locales. Pour assurer la cohérence globale, la mise à jour des croyances dans le RB local d'un agent se fait par la transmission d'observations probabilistes (soft evidence) des variables partagées. En effet, les croyances sur chaque variable partagée O , qui sont les probabilités marginales a posteriori, sont les mêmes pour tous les agents. La méthode de mise à jour de la distribution de probabilité d'un agent utilisant les observations probabilistes consiste à introduire des variables d'observation dans son RB comme le montre la figure 3.8. Par conséquent, chaque agent recevant des messages des autres agents obtient une observation probabiliste pour une ou plusieurs variables d'observation créées de la façon suivante :

1. on crée une variable d'observation Obs pour chaque observation probabiliste. Chaque état de la variable ajoutée correspond aux résultats possibles de l'observation probabiliste. Cette variable appartient à l'agent de type demandeur.

2. on ajoute des arcs dirigés vers Obs à partir de toutes les variables du RB qui ont une influence directe sur l'observation. L'ensemble des parents de Obs d-sépare Obs du reste du RB.
3. on modélise la dépendance de la variable Obs en spécifiant sa table de probabilités conditionnelles $P(Obs|pa(Obs))$ sachant que $P(Obs = o|pa(Obs) = x) = 1$ ssi $x = o$ avec o est la valeur de l'observation.

La procédure de mise à jour par observations probabilistes est détaillée dans [BV02].

Le modèle de [Oud+10] fait également partie de la deuxième catégorie. C'est un modèle qui est basé sur le MSBN de [Xia00] mais qui rajoute une nouvelle propriété appelée limites de Markov ("*Markov boundaries*") pour définir les relations d'indépendances associés à chaque variable (voir annexe A). Cette propriété est utilisée pour éviter l'utilisation d'une autre structure pour l'inférence [Oud+10]. Un RB local, appelé module RB, est associé à chaque sous-domaine. Dans la structure bayésienne modulaire de [Oud+10], une inférence collaborative consiste à partager des croyances (probabilité a posteriori) sur une variable commune entre deux modules bayésiens. Ces deux modules bayésiens ont établi un contrat de partage de croyances ("*Belief Sharing Contract*"). Pour établir une inférence globalement correcte et cohérente dans un RB modulaire, [Oud+10] a fait appel au graphe de cliques (*clusters*). En effet, un RB modulaire correspond à un graphe de cliques dans les travaux de [Oud+10].

Un graphe de cliques $G = (C, S)$ sert à examiner les dépendances entre les modules bayésiens. Soit $C = \{C_1, C_2, \dots, C_n\}$ l'ensemble de cliques contenant chacun un ensemble de variables du RB modulaire. Soit $S = \{S_{i,j} | 1 \leq i \leq n, 1 \leq j \leq n, i \neq j\}$ l'ensemble de tous les séparateurs entre les cliques tel que $S_{i,j} = C_i \cap C_j \neq \emptyset$, $C_i \in C$ et $C_j \in C$. Si l'ensemble S ne contient que les variables communes qui sont impliquées dans le contrat de partage de croyances, C est aussi le graphe d'inférence. Chaque agent d'inférence contient des variables d'entrée R_i et des variables de sortie L_i . L'inférence locale dans chaque module bayésien se déclenche quand une nouvelle évidence se produit au niveau d'une variable L_i . Chaque agent A_i communique par la suite les probabilités a posteriori via les variables R_i à un autre agent A_j , avec $A_i \neq A_j$, par le biais de ses variables L_j . Les agents doivent alors être organisés dans une structure bien définie pour faire passer les probabilités marginales des R_i aux L_j . Le modèle de [Oud+10] a alors divisé le RB monolithique en plusieurs RBs modulaires structurés en différents niveaux pour une inférence qui s'exécute du niveau le plus bas au niveau le plus haut. Cette inférence hiérarchique rajoute des contraintes et ne convient pas à la modélisation des environnements dynamiques dont la structure évolue au cours du temps.

Il existe encore d'autres modèles bayésiens modulaires comme celui de [HC06] qui appartient à la deuxième catégorie de RBs distribués où les modules bayésiens sont organisés dans une structure d'arbre de jonction. Les inférences locales se produisent parallèlement pour réduire le temps d'inférence. Mais cela ne permet pas d'avoir des modules bayésiens indépendants car le système global doit connaître toutes les relations de dépendances entre

les modules bayésiens.

Les RBs distribués de la deuxième catégorie ne permettent pas de représenter la dynamique des systèmes qui évoluent au cours du temps. La **troisième catégorie** peut résoudre partiellement ce problème, comme le modèle de [HC09], mais la modélisation reste toujours statique. Dans le modèle de [HC09], les modules bayésiens transmettent les probabilités marginales via un lien virtuel (les arcs rouges) comme le montre la figure 3.9. L'inférence

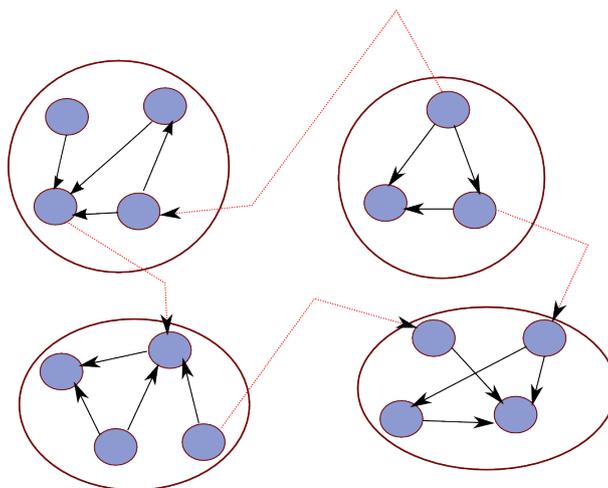


FIGURE 3.9 – Modules RBs liés virtuellement

dans un tel modèle est une inférence à deux étapes en respectant la relation de co-causalité entre les modules bayésiens. Les modules bayésiens infèrent localement l'ensemble des observations certaines dans un premier temps. Le résultat de la première inférence (probabilités marginales a posteriori) est communiqué entre les modules bayésiens sous forme d'observations probabilistes sur les nœuds communs. Ce passage d'observations probabilistes consiste à ajouter un nœud virtuel en réglant sa table de probabilités conditionnelles afin d'appliquer les évidences comme le montre la figure 3.10. Pour un nœud non racine X_i , un nœud X'_i est rajouté et associé à X_i comme un nœud fils. Quand X_i reçoit une observation probabiliste p , la table de probabilités conditionnelles de X'_i est alors mise à jour comme le montre la figure 3.10. Pour un nœud racine X''_i qui reçoit une évidence probabiliste p , c'est la probabilité marginale a priori de X''_i qui est modifiée.

Une construction dynamique des RBs modulaires qui permet l'extension des modèles bayésiens doit être envisagée. C'est le modèle de [LYC15], [LLC16] que nous trouvons le plus pertinent pour modéliser la dynamique d'un SCR qui change de mode fonctionnement pour s'adapter à son environnement.

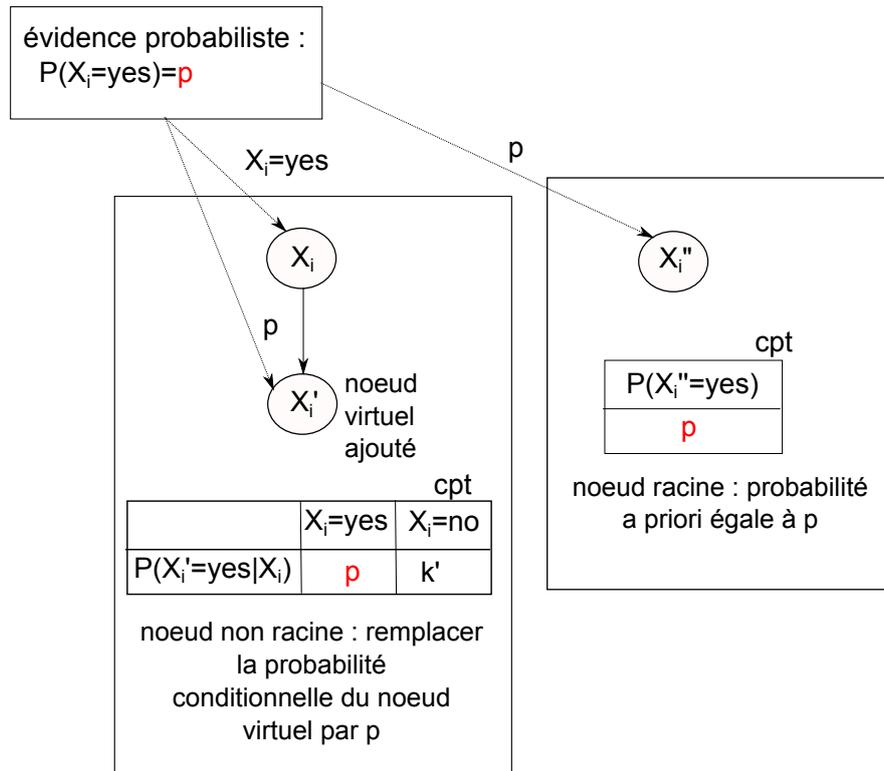


FIGURE 3.10 – Inférence par lien virtuel

3.5.2 Réseaux bayésiens modulaires

Le modèle bayésien modulaire de [LYC15], [LLC16] est la meilleure solution pour un système dont la dynamique évolue et change au cours du temps. C'est une structure bayésienne qui garantit l'extensibilité et l'évolutivité. Ce réseau bayésien modulaire est conçu pour les environnements mobiles. La division d'un RB monolithique en plusieurs modules bayésiens possèdent plusieurs avantages [LLC16] :

- réduire la charge de calcul et l'utilisation de la mémoire sur les serveurs parce que les informations sont pré-traitées dans des agents intelligents à distance pour une inférence modulaire et puis pour une inférence finale en communiquant le résultat des inférences locales.
- réduire le temps d'inférence global en contrôlant le passage d'informations uniquement vers les modules pertinents en utilisant la propriété de d-séparation intermodulaire et l'inférence sélective que nous allons détailler.
- remplacer les modules bayésiens par d'autres et les réutiliser dans d'autres contextes.
- adaptabilité, extensibilité et évolutivité selon la dynamique du système et son environnement.

La modularisation d'un RB monolithique consiste à définir des modules RBs en sélection-

nant les variables correspondant à chaque sous-domaine. Un sous-graphe acyclique est alors construit pour chaque module bayésien en se basant sur le graphe acyclique global du RB monolithique : un module bayésien est un sous-graphe du RB monolithique. Chaque module bayésien partage des variables (des nœuds) avec d'autres sous RBs. Les distributions de probabilités de chaque module bayésien correspondent à celles du RB monolithique.

Définition 3.1

Un module bayésien $\Psi_i = (G_i, P_i)$ est associé à un RB de graphe acyclique $G_i = (V_i, E_i)$ où V_i est l'ensemble des variables, E_i représente l'ensemble des arcs orientés et P_i est la distribution de probabilités dans le module Ψ_i . Chaque module bayésien est défini ou conçu pour un certain objectif en représentant des problématiques d'un domaine bien défini.

Définition 3.2

Un RB modulaire Ω est défini par [LLC16] :

- un couple (Ψ, R) où Ψ représente l'ensemble des modules bayésiens dans Ω et R est l'ensemble de liens de causalités entre ces modules.
- deux modules $\Psi_i = ((V_i, E_i), P_i)$ et $\Psi_j = ((V_j, E_j), P_j)$ ont un lien de causalité s'il existe un lien $R_{i,j} = \{(\Psi_i, \Psi_j) | i \neq j, V_i \cap V_j \neq \emptyset\}$ qui est une relation de cause-à-effet entre ces deux modules bayésiens.

La propriété de d-séparation inter-modulaire est similaire au concept de d-séparation entre des variables/nœuds dans un RB. Il existe également trois types de d-séparation inter-modulaires comme le montre la figure 3.11 : série, convergente et divergente. Deux modules Ψ_i et Ψ_j sont d-séparés sachant un ensemble d'évidences E , que l'on représente par $\Psi_i \perp_m \Psi_j | E$ [LLC16]. Cette d-séparation inter-modulaire implique que tout changement dans Ψ_i ou Ψ_j n'affecte pas l'autre module connecté. Dans le cas d'une connexion divergente ou en série, Ψ_i et Ψ_j sont d-séparés si et seulement si une évidence certaine est appliquée sur le nœud partagé S tel que $\Psi_i \perp_m \Psi_j | S$. Quant à la connexion convergente, Ψ_i et Ψ_j sont d-séparés si S ne reçoit pas d'évidences certaines. Deux modules connectés peuvent avoir plusieurs types d'échanges, cela dépend du type d'évidence. Si deux modules Ψ_i et Ψ_j sont d-séparés, il n'est plus nécessaire de ré-inférer l'autre module si un des deux modules est mis à jour.

3.5.2.1 Inférence modulaire

L'inférence exacte dans un RB monolithique (une seule structure connectée) consiste à faire passer des messages entre les variables du RB comme cela a été détaillé dans la section 3.4. Cependant, le RB modulaire possède des RBs séparés qui sont interconnectés via des nœuds communs. Dans une structure bayésienne modulaire, chaque module RB doit collecter les résultats d'inférence des autres modules qui lui sont connectés. La méthode

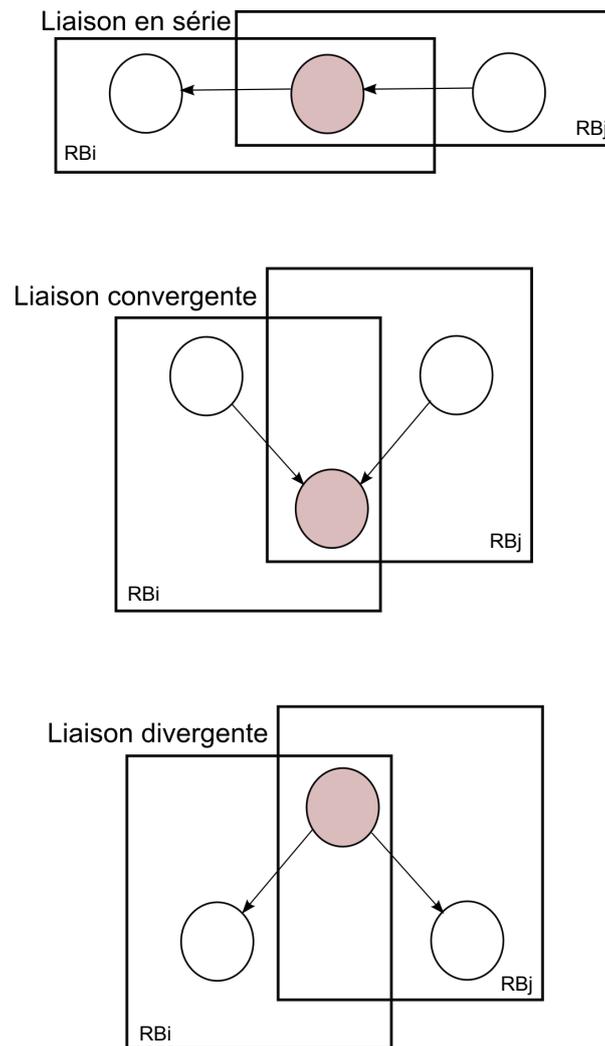


FIGURE 3.11 – Dépendance inter-modulaire [LLC16]

de liens virtuels va servir à transmettre les messages (les évidences incertaines) entre les différents modules connectés. Deux modules bayésiens sont virtuellement connectés s'ils partagent une variable S , appelée S_i dans Ψ_i et S_j dans Ψ_j . Le résultat de l'inférence locale $Bel(S_i) = p_1, p_2, \dots, p_n$ dans le module Ψ_i est propagé dans l'autre module Ψ_j en mettant à jour les probabilités de S_j selon $Bel(S_i)$, où $Bel(S_i)$ est la probabilité a posteriori de S_i après l'inférence dans Ψ_i . Cela revient à mettre à jour la table de probabilités conditionnelles du nœud virtuel comme le montre la figure 3.12. Un nœud virtuel V ne fait pas partie de la construction initiale des modules bayésiens. C'est un nœud auxiliaire utilisé pour pouvoir appliquer des observations incertaines comme cela a été détaillé dans la sous-section 3.4.2.1. Le nœud virtuel V est associé au nœud partagé S comme étant son nœud fils : les probabilités de V sont définies et calculées selon les nouvelles probabilités cibles

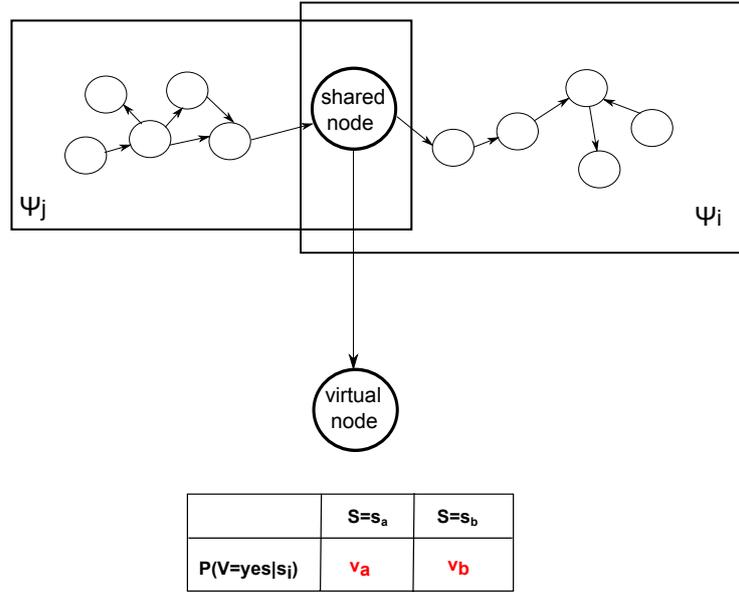


FIGURE 3.12 – Méthode de liens virtuels [LLC16]

de son parent S . Il est caractérisé par deux états : *oui* et *non* et une évidence certaine $P(V = yes) = 1$. Les probabilités de V , $P(V = yes|s_i) = \{v_a, v_b\}$ sont calculées selon l'équation 3.6 de rapport de vraisemblances :

$$\frac{v_a}{v_b} = \frac{P(S = s_a|V = yes)P(S = s_b)}{P(S = s_b|V = yes)P(S = s_a)} \quad (3.6)$$

où $P(S = s_a|V = yes)$ et $P(S = s_b|V = yes)$ représentent les valeurs des évidences incertaines à appliquer au nœud S , et $P(S = s_a)$ et $P(S = s_b)$ sont les probabilités marginales initiales de S . En effet, quand le nœud virtuel V reçoit une évidence certaine, les probabilités a posteriori de S sont définies dans les équations 3.7 et 3.8 en utilisant le théorème de Bayes :

$$P(S = s_a|V = yes) = \frac{P(V = yes|S = s_a)P(S = s_a)}{P(V = yes)} \quad (3.7)$$

$$P(S = s_b|V = yes) = \frac{P(V = yes|S = s_b)P(S = s_b)}{P(V = yes)} \quad (3.8)$$

Le rapport des deux équations $\frac{3.7}{3.8}$ donne l'équation 3.9 :

$$\frac{P(S = s_a|V = yes)}{P(S = s_b|V = yes)} = \frac{P(V = yes|S = s_a)P(S = s_a)}{P(V = yes|S = s_b)P(S = s_b)} \quad (3.9)$$

où $P(S = s_a|V = yes)$ et $P(S = s_b|V = yes)$ représentent les valeurs des évidences incertaines à appliquer sur le nœud S .

Pour calculer finalement les probabilités du nœud virtuel, nous transformons l'équation 3.9 en équation 3.10, ce qui correspond à l'équation 3.6 :

$$\frac{P(V = yes|S = s_a)}{P(V = yes|S = s_b)} = \frac{P(S = s_a|V = yes)P(S = s_b)}{P(S = s_b|V = yes)P(S = s_a)} \quad (3.10)$$

3.5.2.2 Inférence sélective

L'inférence sélective va permettre d'optimiser le temps d'inférence dans le RB modulaire et d'avoir des résultats d'inférence fiables. L'algorithme d'inférence sélective est décrit en détail dans [LLC16]. Il commence par choisir les nœuds cibles dont le RB modulaire a besoin de connaître les états en se basant sur l'ensemble des évidences collectées. Si un module contient un nœud cible, il devient également un module cible. Les modules cibles possèdent la plus haute priorité, ce seront les derniers modules sur lesquels la procédure d'inférence sera appliquée. Les modules qui ont des relations de dépendances avec les modules cibles ont le deuxième niveau de priorité. Les modules qui sont connectés aux modules de deuxième priorité ont le niveau d'après. Après avoir sélectionné la séquence d'inférence de modules bayésiens, la deuxième étape de l'inférence sélective consiste à vérifier la d-séparation entre les modules bayésiens pour éviter de ré-inférer des modules qui sont d-séparés des autres modules qui ont reçus des changements. Cette méthode permet de propager les messages dans le RB modulaire comme dans le RB monolithique mais avec un coût d'inférence réduit [LLC16].

3.6 Réseaux bayésiens pour le diagnostic

Le diagnostic d'un système consiste à localiser une défaillance afin de prendre une décision rapide pour remettre le système dans son état normal. Le diagnostic d'un défaut consiste à utiliser une méthode permettant de déterminer si un composant est défaillant ou non et de localiser l'origine de la défaillance à partir des observations. En outre, une défaillance est un état qui peut mener à un dysfonctionnement du système. Un défaut peut être caractérisé par une déviation, en dehors d'un intervalle acceptable, d'une variable ou d'un paramètre observé. La tâche de diagnostic est divisée en trois sous-tâches [CP12] :

- **détection du défaut** : elle consiste à indiquer et à observer un dysfonctionnement dans le système.
- **localisation du défaut** : elle consiste à déterminer l'élément défaillant.
- **identification du défaut** : elle consiste à déterminer la forme et l'amplitude du défaut.

Les premières méthodes de diagnostic ont été basées sur la redondance de matériels considérés critiques pour le fonctionnement d'un système [MMZ08]. Cependant, cette méthode

possède des inconvénients liés aux coûts dus à la multiplication des éléments et au poids supplémentaire. D'autres méthodes du domaine de l'automatique et de l'intelligence artificielle ont vu le jour permettant d'éliminer la redondance matérielle. Les méthodes de diagnostic se partagent en deux approches [Ise05] :

- l'approche quantitative utilisant des modèles analytiques tels que l'espace de parité, les observateurs et l'estimation de paramètres, les bond graphs...
- l'approche qualitative qui regroupe des méthodes basées sur les techniques de l'intelligence artificielle et qui ne font pas appel aux modèles analytiques.

Dans la première approche, il faut avoir un modèle mathématique qui représente les phénomènes physiques pour réaliser le diagnostic. Cependant, il est parfois très difficile d'avoir le modèle qui traduit fidèlement le comportement du système. En ce qui concerne la deuxième approche, elle exploite les données disponibles pour faire l'apprentissage et l'expertise du système pour construire les règles. Mais dans le cas où la base de données est incomplète ou si plusieurs cas sont indisponibles, il sera difficile d'utiliser cette approche. Dans le cas de systèmes complexes pour lesquels des connaissances expertes sont disponibles, il est possible d'utiliser les réseaux bayésiens [Naï+07].

Plusieurs modèles de réseaux bayésiens ont été proposés pour faire le diagnostic. Pour les systèmes dont le comportement est aléatoire, le RB, en effet, est utilisé pour évaluer l'état du système. Le réseau bayésien est un bon formalisme de modélisation dans l'incertain [WS13], [WP17]. Un autre avantage est la possibilité de représenter les dépendances entre les défaillances des composants d'une part et entre les composants et leurs valeurs de sortie d'autre part sous forme de tables de probabilités conditionnelles. Une telle structure permet de surveiller l'état des composants du système. Les observations vont renseigner sur l'état de santé d'un composant et les probabilités a posteriori, calculées grâce à l'algorithme d'inférence du RB, vont donner l'état le plus probable : normal, dégradé ou défectueux.

[Bot+08] ont développé un réseau bayésien pour le diagnostic d'un satellite. Le système est composé d'amplificateurs, de convertisseurs de signaux, de modems, de routeurs qui sont susceptibles de tomber en panne. Ils proposent dans leur modèle pour le diagnostic deux types de nœuds : un nœud mesurable qui correspond à l'état de paramètres mesurables et un nœud non mesurable *failure node* qui représente le défaut d'un composant ou de tout le système. Chaque nœud *failure* possède deux états : *fail* qui est un état défaillant ou *no fail* qui représente un système ou composant en bon état. Un autre type de nœud peut être utilisé dans le modèle, il représente une variable d'environnement dans lequel le système fonctionne. Le diagnostic dans ce type de modèle consiste à déterminer la probabilité marginale maximale des nœuds non mesurables sachant l'état de nœuds mesurables, ce qui donne comme résultat la cause la plus probable de la défaillance du système. Cela revient à déterminer la probabilité de l'état *fail* d'un composant ou du système pendant une période de temps fixe. Cette probabilité dépend des évidences au niveau des nœuds mesurables. Le nœud mesurable, dans leur modèle, est le parent d'un seul type de défaut au niveau d'un composant. Le nœud *System failure* possède quatre nœuds parents qui correspondent

à la défaillance possible des quatre composants du système (amplificateur, convertisseur du signal, modem, routeur). Le système est en état défaillant si l'un de ses composants est défaillant. Par conséquent, le système n'est dans l'état non défaillant que lorsque tous ses composants sont dans l'état non défaillant.

[CXY13] proposent deux types de RB : un RB statique pour le diagnostic d'un train à grande vitesse et un réseau bayésien dynamique (RBD) pour la maintenance. Leur objectif est de diagnostiquer la cause de l'arrêt d'un train à grande vitesse et de prendre la décision de maintenance pour améliorer l'état du système dans le futur. Après avoir déterminé la cause de la défaillance du système par le RB statique, ils utilisent un RBD pour la prise de décision de maintenance. Dans leur modèle dynamique (réseau bayésien dynamique pour la maintenance de l'*ATPCU* (*Automatic Train Protection Control Unit*), ils utilisent un nœud monitor qui a pour rôle la surveillance. Quand le nœud monitor découvre une défaillance au niveau du système, il communique l'information au nœud *maintenance* (la probabilité de la maintenance augmente) pour décider d'effectuer ou non une action de maintenance. Le nœud maintenance agit sur le système *Automatic train protection* (ATP), qui aide à éviter les collisions, et améliore l'état du nœud *ATPCU* à l'instant suivant. Le nœud Stop, qui représente l'état du train, possède 2 états : Yes et No. Lorsque le train est en arrêt, l'*ATPCU* est en état dégradé. Le nœud *System condition* représente les conditions dans lesquelles le système fonctionne. En effet, tout changement dans l'environnement a un impact sur le système. Si l'état du nœud *System condition* est bon, l'état de l'*ATPCU* sera en bon état sinon il va se dégrader.

[SMM12] présente un modèle bayésien qui a été développé par une équipe de la NASA pour le *Software Health Management (SWHM)* pour détecter et diagnostiquer les défaillances d'origine logicielle ou bien matérielle. Ils ont appliqué leur modèle sur un robot explorateur et sur un système de contrôle d'avion. L'approche bayésienne utilisée est une approche modulaire, hiérarchique et réutilisable. Leur modèle se compose de différents types de nœuds discrets comme le montre la figure 3.13 :

- **nœud Commande C** : ce nœud représente une vérité terrain et indique une commande, une action, un mode ou bien d'autres états connus. C'est un nœud observable.
- **nœud Sensor S** : ce nœud est aussi observable. Il représente, par exemple, les mesures obtenues à partir de la surveillance du logiciel et du matériel.
- **nœud Health H_S , H_U** : ce nœud (non mesurable ou observable) représente l'état d'un composant ou d'un capteur. L'état de ce nœud est la sortie du modèle.
- **nœud Status U** : ce nœud représente l'état non observable d'un composant logiciel ou matériel ou l'état du sous-système.

Les relations de cause à effet dans ce modèle sont les suivantes :

- $C \rightarrow U$: cette relation explique comment l'état du système est influencé (change) par la commande appliquée.
- $H_S \rightarrow S$: ce lien représente l'influence de l'état de santé du système sur le capteur

qui fournit les mesures d'observations du système.

- $U \rightarrow S$: ce lien représente l'influence de l'état du système sur le capteur.

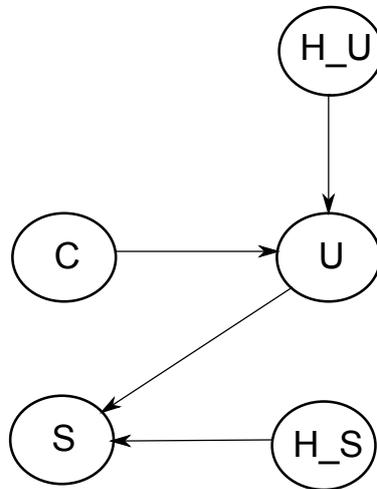


FIGURE 3.13 – Relations possibles entre les différents types de nœuds du SWHM [SMM12]

[PC07] ont développé un modèle bayésien pour le diagnostic et le pronostic qui a une structure en couches comme le montre la figure 3.14 :

- **couche "components"** : une couche de nœuds composants ou modes de défaillances des composants dont les états sont le résultat du diagnostic.
- **couche "health observations"** : une couche de nœuds d'observations sur l'état des composants.
- **couche "sub-systems"** : une couche de nœuds qui représentent l'état du sous-système.
- **couche "usage"** : une couche de nœuds observables qui représentent l'usage du système dans le présent (dans le cas du diagnostic) et dans le futur (dans le cas du pronostic).

Une telle structure permet de réduire le nombre de paramètres nécessaires pour définir le RB. L'objectif est de définir le diagnostic des composants à partir des observations et des usages. La méthode de [PC07] consiste à décomposer le système en sous-systèmes et à les modéliser séparément. Finalement, on intègre les sous-modèles en un seul modèle global. Pour construire un tel modèle il faut :

- décomposer le système en sous-systèmes.
- définir les composants du système/sous-systèmes.
- définir les observations disponibles (tests, symptômes, messages d'erreur).
- définir une liste de défaillances des composants.
- faire correspondre les observations pertinentes aux défaillances.

Les étapes du raisonnement dans ce modèle bayésien pour le diagnostic et le pronostic ont été décrites dans [PC07] :

- rassembler les données de défaillances des composants à partir des simulations et

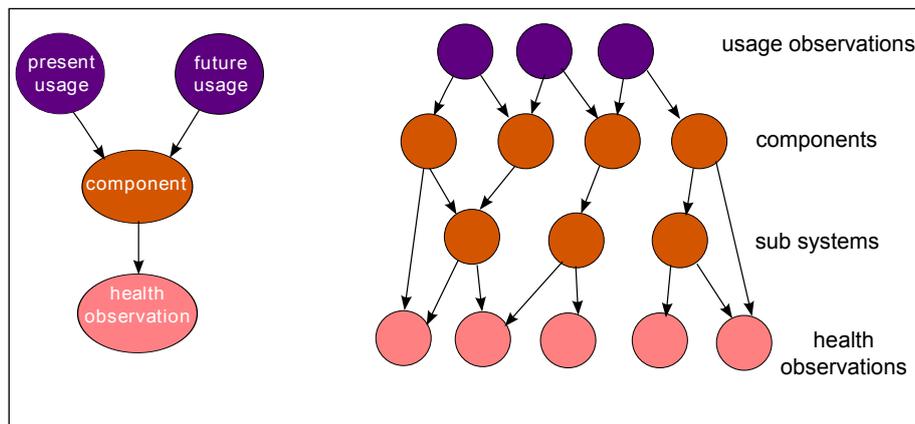


FIGURE 3.14 – Modèle bayésien en couches de diagnostic de [PC07]

des tests.

- créer des histogrammes de défaillances du composant en fonction de l’usage du composant c_i et en fonction des observations o_i . Le premier histogramme représente la densité de défaillance f_{c_i} et le deuxième représente celle des observations f_{o_i} .
- pour effectuer le diagnostic, calculer la probabilité de défaillance d’un composant sachant l’usage actuel c_i et l’observation actuelle o_i .
- pour faire le pronostic avec le même modèle, utiliser l’usage actuel c_i , l’usage futur c_j et la tendance de l’observation o_k .

Le résultat du diagnostic est celui du calcul de la probabilité de $P(c_{def}|up_i, uf_i, o_k)$ dans le RB où : c_{def} est l’état défaillant du composant C ; up_i est l’usage actuel du composant qui correspond à c_i ; uf_i est l’usage futur du composant ; o_k est l’observation et $up_i = uf_i$. Dans le cas du pronostic, on prend $i < j$ où uf_j est un usage plus long dans le temps que l’usage actuel up_i .

Le modèle bayésien pour le diagnostic d’un système commandé en réseau que nous allons présenter par la suite est inspiré de ces RBs pour le diagnostic.

3.7 Réseaux bayésiens pour le diagnostic distribué

Le contrôle distribué présente non seulement plusieurs avantages mais aussi des défis en terme de diagnostic et de surveillance. Les RBs monolithiques sont par ailleurs limités par une structure et une inférence centralisée. Pour un système distribué, tel que le contrôle distribué, l’utilisation des RBs monolithiques pour le diagnostic n’est pas aussi intéressant et efficace que les RBs distribués [XG99]. Un RB distribué fournit un bon support pour la problématique de diagnostic distribué : connaissances et inférence distribuées. Il est aussi plus adéquat pour modéliser un système reconfigurable. Un RB distribué a la capacité de

diagnostiquer les défauts le plus rapidement possible et permet par conséquent une maintenance le plus tôt possible et un temps de panne plus court. La surveillance et le diagnostic des systèmes complexes pendant leur fonctionnement implique que le système doit s'adapter aux changements et à l'environnement. Le diagnostic modulaire s'avère être une bonne solution pour les systèmes complexes et distribués. Il n'existe pas beaucoup de travaux qui utilisent les RB distribués et modulaires pour le diagnostic, la plupart les ont introduits sous forme d'une structure centralisée. La dynamique des systèmes modulaires, où des modules se rajoutent ou s'enlèvent, fait que l'utilisation de l'approche centralisée est difficile pour gérer des changements et des reconfigurations rapides. En outre, l'effort de conception dans une telle architecture de diagnostic modulaire est optimisé vu que chaque RB local est modélisé par un expert du domaine qui lui correspond. [SL13] ont proposé un modèle bayésien multi-agents : un MSBN pour le diagnostic distribué d'un système d'assemblage modulaire qui est un système distribué et reconfigurable. Le MSBN a permis d'avoir des sous-modèles bayésiens, encapsulés dans des agents intelligents, pour le diagnostic local où chaque sous-modèle correspond à un domaine bien défini. Ces sous-modèles bayésiens sont ensuite intégrés dans une procédure globale de diagnostic où les agents de diagnostic échangent des informations pour un raisonnement plus global [SL13]. [XG99] ont appliqué le MSBN dans le contexte de surveillance et de diagnostic des équipements "*monitoring and diagnosis*" M&D sur plusieurs types de systèmes tel que le diagnostic basé sur internet "*Internet based Diagnosis*". Chaque sous RB est l'unité centrale d'un agent autonome de M&D. [ATC11] ont développé un système multi-agent pour le diagnostic de défauts dans un réseau HAN, dans le cadre d'un projet appelé *MAGNETO*, pour gérer un ensemble de foyers à distance. C'est un modèle bayésien distribué qui sert à diagnostiquer les causes de défaillance du réseau et des problèmes du service. Le RB pour le diagnostic est divisé en plusieurs sous RBs : des agents bayésiens pour diagnostiquer les erreurs du réseau HAN, d'autres pour les erreurs de connectivité dans les fournisseurs d'accès à internet FAI (*ISP : Internet Service Provider*). Les agents bayésiens communiquent en utilisant les évidences virtuelles pour une inférence distribuée entre différents domaines.

Pour un système commandé en réseau, tel que le robot mobile commandé par une station distante, la station et le robot n'ont pas la même vision de l'environnement ou de l'état du système global. En outre, le robot peut être dans un premier temps commandé par la station ou bien autonome et dans ce cas l'architecture de commande, voire l'architecture de diagnostic doit être mise à jour pour s'adapter à la dynamique du système et de l'environnement. Une architecture de diagnostic modulaire s'avère être une bonne solution pour le diagnostic de ce système commandé en réseau. Un modèle bayésien modulaire pour le diagnostic, ayant la même structure et les mêmes caractéristiques que celles du modèle de [LLC16], et tel que chaque module bayésien possède une architecture en couches inspirée du modèle de [PC07] est développé pour répondre à la problématique de diagnostic des systèmes commandés en réseau.

3.8 Conclusion

Les RBs sont un outil de raisonnement dans l'incertain qui permettent de représenter des relations de dépendances cause-à-effet. L'objectif de l'utilisation des RBs peut être le diagnostic, le pronostic ou bien la combinaison des deux tel que dans le modèle de Przytula [PC07] qui sera la base de notre modélisation bayésienne pour le diagnostic d'un SCR sans fil : un robot mobile commandé en réseau par une station distante. Les réseaux sans fil sont des sources de comportement indéterministe et peuvent engendrer des dysfonctionnements aléatoires du système. La communication peut être non fiable à cause de pertes de données, de retard, de dégradation de la qualité de la connectivité. Pour les SCRs sans fil, une méthode probabiliste telle que les RBs s'avère alors être une bonne solution. En outre, les SCRs sans fil peuvent changer d'architecture en cas de rupture de connexion entre le robot et la station, dans ce cas le robot devient autonome et le réseau de communication n'intervient pas dans la boucle de commande du système. Une reconfiguration doit être envisageable. Finalement, les capacités de calcul et de mémoire sont limitées au niveau du robot et les ressources de communication sont parfois indisponibles, d'où l'intérêt de la modélisation bayésienne modulaire pour le diagnostic que nous allons détailler dans le chapitre 5.

Performance et qualité de service dans les réseaux sans fil

Sommaire

4.1	Introduction	55
4.2	Contraintes temps réel des SCRs	56
4.3	Analyse de la performance de la QoS pour les systèmes commandés en réseau sans fil	57
4.4	Modèle analytique du DCF dans les réseaux IEEE 802.11	58
4.4.1	Chaînes de Markov	58
4.4.2	Théorie des files d'attente	60
4.4.3	Définition du comportement stochastique	60
4.4.4	Étude probabiliste du modèle markovien des réseaux IEEE 802.11 DCF	63
4.5	Définition des paramètres de la QoS	68
4.5.1	Débit de saturation et période d'échantillonnage critique	68
4.5.2	Retard moyen aller-retour	69
4.5.3	Taux de pertes de paquets	70
4.5.4	Gigue	70
4.6	Cas d'études	71
4.7	Conclusion	73

4.1 Introduction

Le lien de cause à effet entre la QoS et la QoC implique que dans le cas où le réseau sans fil n'assure pas le niveau de QoS requis (un état défailant du réseau), la QoC et la performance du système deviennent dégradées. Cette influence est due au délai, aux pertes de paquets, à la gigue et au débit comme cela a été détaillé dans le chapitre 2. Le niveau requis de QoS doit être défini selon ces paramètres dans le but de garantir une bonne

QoS. Les Wlans IEEE 802.11 sont utilisés dans plusieurs applications de SCRs sans fil surtout en robotique mobile. Le mécanisme d'accès au médium pour ce type de réseaux est le DCF. C'est une procédure, basée sur le CSMA/CA, qui sert à contrôler l'accès à un canal de communication partagé entre plusieurs stations communicantes comme cela a été expliqué dans le chapitre 2. Dans ce chapitre, nous proposons d'étudier les performances que le DCF peut fournir au réseau des SCRs sans fil pour déterminer la QoS requise en matière de délai moyen, de taux de pertes de paquets, de la valeur de la gigue et de débit de saturation. Les études qui ont été faites sur le DCF de l'IEEE802.11 utilisent la théorie des chaînes de Markov pour modéliser le comportement stochastique des réseaux sans fil. Le but de notre étude est d'ajuster le modèle de [Bia00] (voir figure 2.5 du chapitre 2) pour qu'il représente le comportement périodique du trafic réseau d'une boucle de commande dans des conditions de non saturation i.e, le nœud capteur n'a pas toujours un paquet à transmettre. Le modèle de [Bia00] est le premier modèle stochastique basé sur les chaînes de Markov qui a décrit la procédure d'accès au canal de communication. Comme cela a été détaillé dans la sous-section 2.4.2.2, ce modèle a subi des extensions pour le rendre plus conforme au comportement du protocole de communication utilisé dans les réseaux IEEE 802.11. La chaîne de Markov bi-dimensionnelle de [Bia00] est une bonne solution stochastique pour représenter les différents états de transmission d'un paquet et pour résoudre analytiquement les expressions des variables que nous cherchons à déterminer. Ces variables sont importantes pour déterminer les expressions analytiques des paramètres qui quantifient l'état de la QoS. Le réseau des SCRs sans fil est constitué d'un contrôleur et de n capteurs et par conséquent de n boucles de commande partageant un seul contrôleur implémenté sur une station distante.

4.2 Contraintes temps réel des SCRs

Le trafic réseau d'un SCR sans fil a des exigences et des caractéristiques différentes de celles du service d'un réseau *best effort*. Le SCR sans fil est alors un système temps réel caractérisé par un niveau bien défini de QoS pour lui garantir une bonne performance. Les conditions temps réel du trafic imposent des limites de performance pour les paramètres de la QoS : la bande passante ou le débit, le délai, la gigue et le taux de perte de paquets. Dans un réseau de SCRs sans fil, les capteurs envoient périodiquement le paquet de mesures au contrôleur. Quand le contrôleur reçoit les mesures des capteurs, il calcule les valeurs de la commande et les envoie via le réseau sans fil aux actionneurs. Les actionneurs et les capteurs sont généralement installés sur le même système qui est le robot commandé en réseau dans notre cas.

On considère un SCR sans fil, composé d'un seul contrôleur et de n robots mobiles qui se déplacent dans la zone de couverture. Le réseau est alors occupé par un trafic d'aller retour (round trip) représentant le flux de données de chaque boucle de commande.

Chaque boucle de commande, implémentée via le réseau, est caractérisée par une période d'échantillonnage notée T . Les paquets transmis via le réseau subissent des retards de transmission. Dans les SCRs sans fil, ayant une boucle de commande implémentée via le réseau, le temps d'aller-retour de transmission T_{RTT} (round trip delay) est un paramètre important pour étudier les performances temps réel du système. Le retard T_{RTT} est le délai entre l'instant où le capteur commence la transmission du paquet de mesures et l'instant où les actionneurs reçoivent le paquet de commande. Pour répondre aux exigences temps réel, une condition, décrite par l'inégalité 2.1, doit être vérifiée. En effet, le délai d'aller retour T_{RTT} doit être limité par la valeur de la période d'échantillonnage T .

Réduire la période d'échantillonnage T permet d'avoir une bonne QoS. Cependant, plus la période d'échantillonnage est petite et plus le réseau devient surchargé comme cela a été détaillé dans le chapitre 2. En effet, les petites périodes d'échantillonnage provoquent une instabilité du réseau avec de grands délais et un grand taux de pertes de paquets. A cause du comportement stochastique du réseau sans fil, une méthode stochastique est nécessaire pour représenter au mieux la dynamique des réseaux sans fil et définir l'expression analytique de la plus petite valeur de la période d'échantillonnage. La période d'échantillonnage ne peut pas descendre au-dessous d'une valeur critique T_c . Cette valeur critique garantit la vérification de la condition temps réel décrite en 2.1. Dans le cas où la valeur de la période d'échantillonnage atteint sa valeur critique, le débit dans le réseau atteint sa valeur maximale qu'on appelle *débit de saturation* S . On peut définir également une valeur de la gigue dans le réseau et un taux de perte de paquets acceptable.

4.3 Analyse de la performance de la QoS pour les systèmes commandés en réseau sans fil

Les deux paquets de chaque boucle de commande sont des paquets de petite taille fixe. Pour définir les valeurs critiques de chaque paramètre de la QoS, il faut tout d'abord définir la plus petite période d'échantillonnage (ou sa valeur critique) qui garantit la stabilité de la QoS pour un certain nombre n de boucles de commande. Dans ce cas, le débit dans le réseau va atteindre sa valeur maximale qu'on appelle *débit de saturation* S et le retard d'aller retour moyen T_{avgRTT} atteint son minimum. De plus, il faut également définir les valeurs de la gigue et du taux de pertes de paquets acceptables qui correspondent à T_c . Le modèle de Markov proposé représente le trafic réseau de chaque boucle de commande. Pour la transmission du paquet de mesures, on ajoute au modèle de [Bia00], une première étape qui représente la condition de non saturation ou une file d'attente vide car le capteur n'envoie qu'à chaque période d'échantillonnage T comme cela a été décrit dans le travail de [TCT16]. Par conséquent, on ne trouve pas toujours un paquet de mesures en tête de la file d'attente du capteur. Les capteurs fournissent des informations sur la sortie du

système à chaque instant kT . Le contrôleur, quant à lui, est implémenté sur un point d'accès et partagé par n boucles de commande, et la tâche contrôleur ne se déclenche que lorsque le paquet de mesures associé est reçu. On rajoute alors un autre ensemble d'étapes de backoff pour le paquet de commande. Ajouter le deuxième ensemble d'étapes de backoff est pertinent pour représenter la synchronisation entre la transmission du paquet de mesures et du paquet de commande correspondant. Cette extension nous permet d'étudier les caractéristiques du trafic temps réel de chaque boucle de contrôle. En outre, comme le point d'accès est partagé entre plusieurs boucles de commande, sa file d'attente n'est pas toujours vide. Le paquet de commande peut ne pas accéder directement au début de la file d'attente et par conséquent il peut ne pas commencer immédiatement la procédure de DCF pour être transmis. On ajoute alors un état de mise en file d'attente avant les étapes de backoff du paquet de commande.

4.4 Modèle analytique du DCF dans les réseaux IEEE 802.11

4.4.1 Chaînes de Markov

Une chaîne de Markov est un processus stochastique défini sur un espace discret ou dénombrable comme une suite de variables aléatoires $(X_n, n \in \mathbb{N})$ et qui modélise l'évolution dynamique d'un système [Jed09]. Elle est caractérisée par une matrice de transition P qui contient les probabilités de transition d'un état à un autre. Le processus markovien possède une propriété fondamentale, la propriété de Markov, qui le décrit comme étant un processus sans mémoire [Jed09], c'est-à-dire que pour déterminer l'état futur du système, il ne faut prendre en compte que l'état actuel. Le futur et le passé sont indépendants conditionnellement au présent.

Soit E un espace discret dénombrable appelé espace d'états. La suite X_n est une chaîne de Markov de loi initiale μ définie dans E si [Jed09] :

- la chaîne vérifie la condition initiale $P(x_0) = \mu(x_0)$
- pour tout $n \in \mathbb{N}$, pour tout (x_0, \dots, x_{n+1}) dans E tels que $P(X_n = x_n, \dots, X_0 = x_0) > 0$,

$$P(X_{n+1} = x_{n+1} | X_n = x_n, \dots, X_0 = x_0) = P(X_{n+1} = x_{n+1} | X_n = x_n) \quad (4.1)$$

- La chaîne est dite homogène lorsque les probabilités de transition ne dépendent pas du temps n , c'est-à-dire pour tout $n \in \mathbb{N}$ et tout x et $y \in E$:

$$P(X_{n+1} = y | X_n = x) = P(X_1 = y | X_0 = x) = P(x, y) \quad (4.2)$$

avec P est la matrice stochastique de transition de la chaîne de Markov.

Définition 4.1

La matrice de transition $P = (P(x, y), x, y \in E)$ est une matrice stochastique dont tous les éléments sont positifs ou nuls et dont la somme des coefficients sur une ligne est égale à 1. Pour tout $x, y \in E$, on a :

$$P(x, y) \geq 0, \quad \sum_{z \in E} P(x, z) = 1$$

Si une chaîne de Markov (X_n) est définie sur un ensemble d'états fini, comme le montre la figure 4.1, alors elle possède au moins une loi de probabilité stationnaire π qui sert à étudier les comportements asymptotiques de cette chaîne de Markov [Jed09].

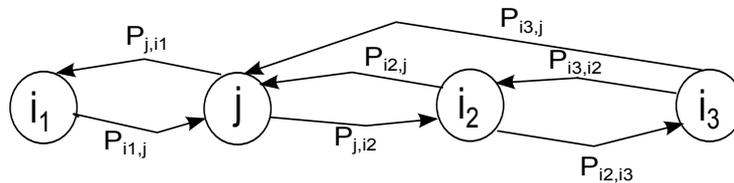


FIGURE 4.1 – Exemple d'une chaîne de Markov

Définition 4.2

Une probabilité π sur E est appelée une probabilité stationnaire (ou invariante) d'une chaîne de Markov à états finis et de matrice de transition P si $\pi = \pi P$ ou de façon équivalente, si pour tout $j \in E$, on a :

$$\begin{aligned} \pi_j &= \sum_{i \in E} \pi_i P(i, j) \\ \sum_{i \in E} \pi_i &= 1 \end{aligned} \quad (4.3)$$

La probabilité $P(i, j)$ correspond à la probabilité de transition de l'état i à l'état j . La probabilité stationnaire π_i , qui sert à étudier les comportements asymptotiques de la chaîne de Markov, permet de définir la loi de X_n qui est la même que celle de X_0 en régime stationnaire. En effet, X_0 représente l'état de la chaîne de Markov correspondant à la loi initiale et le comportement asymptotique d'une chaîne de Markov implique que la loi de X_n devient constante/stationnaire au cours du temps. Le système d'équations 4.3 permet de calculer la distribution de probabilité stationnaire π en lui trouvant des solutions explicites. C'est un outil que nous allons utiliser dans la suite pour résoudre la chaîne de Markov de la figure 4.2 afin de trouver les expressions analytiques des paramètres de la QoS.

4.4.2 Théorie des files d'attente

La théorie des files d'attente traite du problème de gestion des clients qui arrivent auprès d'un serveur, partagé entre eux, pour se faire servir [Jed09]. Dans le cas d'un seul serveur, les clients ne peuvent pas être servis en même temps. Ils doivent attendre dans une file d'attente jusqu'à ce que le serveur soit disponible. Les clients dans la file d'attente du serveur sont ordonnés en premier-entré, premier-sorti (FIFO), s'il n'existe pas une procédure de gestion de priorités dans le serveur ou une procédure d'ordonnement. Un système de file d'attente est caractérisé par des lois qui décrivent les flux des requêtes clientes (les inter-arrivées) qui arrivent au serveur et leur temps de service. La discipline de service décrit le nombre de serveurs et la capacité du système qui est le nombre maximal de clients admis dans la file d'attente. Elle représente aussi la politique de choix du client suivant à servir [Jed09]. Dans la notation de Kendall, une file d'attente est présentée par la notation suivante [Szt12] : $A/B/M/k/n/D$ avec :

- **A** : la fonction de distribution de probabilité du temps d'inter-arrivées.
- **B** : la fonction de distribution de probabilité du temps de service.
- **M** : le nombre de serveurs.
- **k** : la capacité de la file d'attente en termes de nombre de clients comprenant aussi celui qui est en service.
- **n** : le nombre de sources de clients qui existent.
- **D** : la discipline de service (par exemple FIFO).

Pour les variables aléatoires, la densité de fonction de probabilité qui est exponentielle est notée par M . La notation $M/M/1/K$ représente un système dont le temps d'inter-arrivée suit une loi de Poisson, le temps de service, quant à lui, suit une loi exponentielle, et il n'existe qu'un seul serveur à capacité limitée K .

Nous avons précisé précédemment que la file d'attente du contrôleur n'est pas toujours vide. Le contrôleur est considéré comme un serveur de commandes dont la file d'attente est de capacité K . L'état de celle-ci sera modélisé par un état Q dans le modèle markovien de la figure 4.2.

4.4.3 Définition du comportement stochastique

Le modèle de Markov proposé dans la figure 4.2 représente les états de transmissions des deux paquets de chaque boucle de commande dans l'hypothèse d'avoir les conditions idéales pour la transmission (pas d'interférence et pas de terminaux cachés). Dans ce cas, une erreur de transmission est due à une collision de paquets. Soit p la probabilité de collision pour un paquet en transmission telle que p est indépendante de la procédure du backoff. On suppose que le temps du backoff est décrémenté à la fin de chaque intervalle de temps (time slot) [Bia00]. L'état de transmission de chaque paquet est décrit par j et k

tels que j représente l'étape du backoff et k est le temps de backoff qui prend une valeur uniformément répartie dans $[0, W_j - 1]$. La procédure d'accès aléatoire dans les Wlans est présentée par deux processus stochastiques. Soient $b(t)$ le processus aléatoire représentant l'état du temps de backoff dans l'intervalle de temps t (time slot) et $s(t)$ le processus aléatoire correspondant à l'étape du backoff j tel que $j \in [0, R]$ pour la transmission du paquet de mesures et $j \in [R + 1, 2R + 1]$ pour la transmission du paquet de commande. Le processus aléatoire $b(t)$ est uniformément distribué dans l'intervalle $[0, W_j - 1]$ dans la j^{eme} période ou étape du backoff. La chaîne de Markov à temps discret de la figure 4.2 décrit le processus à deux dimensions $\{s(t), b(t)\}$ pour les deux paquets de la boucle de commande. On ajoute l'étape $j = -1$ pour représenter l'état de la file d'attente de

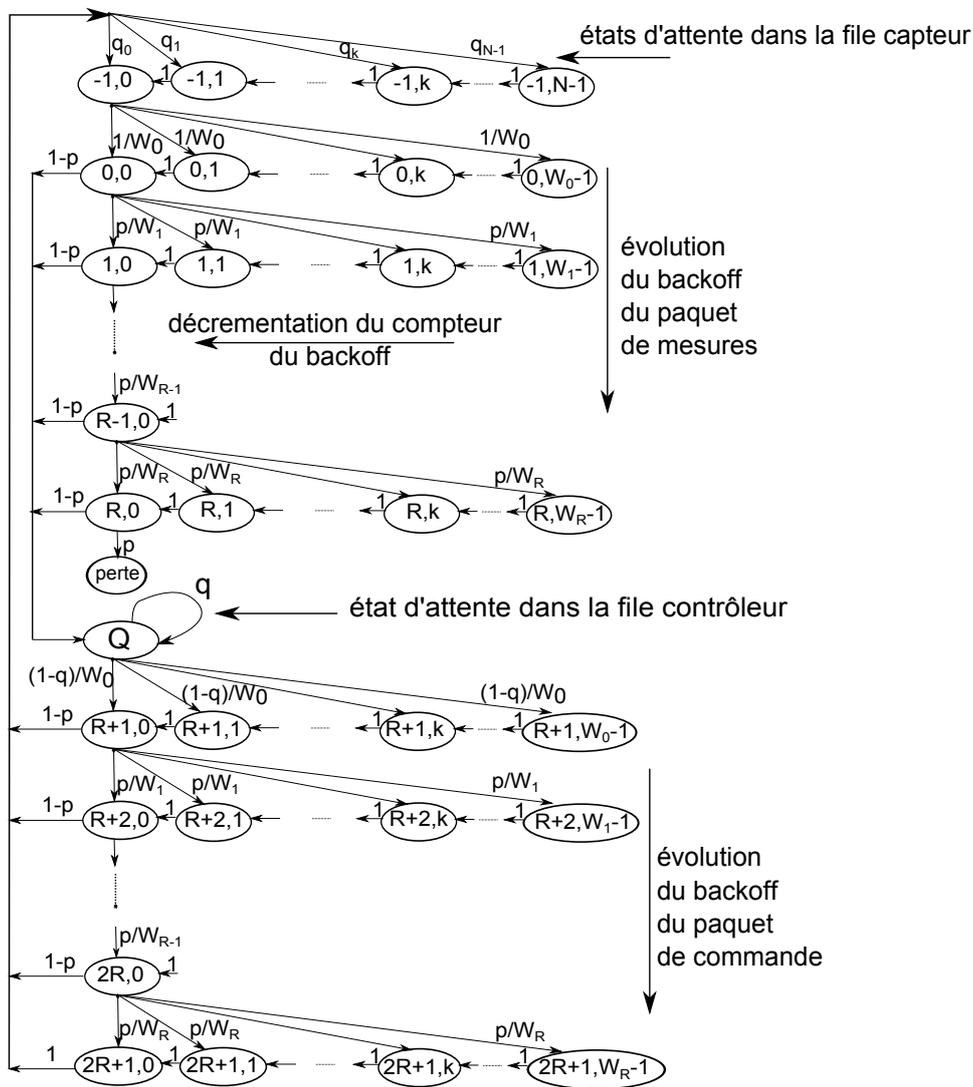


FIGURE 4.2 – Modèle de Markov pour la transmission des paquets de chaque boucle de commande

transmission au niveau du capteur. Cette file d'attente s'inspire des travaux [TCT16]. Dans cette étape, on modélise le temps restant après la transmission des deux paquets de la boucle de commande et avant de commencer une nouvelle période d'accès au canal de communication pour envoyer le paquet de mesures. Ainsi, nous avons N états d'attente $(-1, k)$ dans la file du capteur avec $k \in [0, N - 1]$. Les capteurs fonctionnent en temps discret dont l'unité de temps est le T_{slot} qui représente la durée de la décrémentation dans chaque intervalle de backoff. Le T_{slot} est aussi l'unité de temps dans la chaîne de Markov à temps discret de la figure 4.2. Par conséquent, le nombre N d'états d'attente est défini dans l'équation 4.4 :

$$N = \frac{T - T_{minRTT}}{T_{slot}} \quad (4.4)$$

avec T_{minRTT} le temps d'aller retour minimal.

L'état actuel d'attente $(-1, k)$ avant qu'un paquet de mesures soit prêt pour la transmission est défini à partir du délai d'aller retour de transmission T_{RTT} qui appartient à un intervalle de temps $[T_{minRTT} + iT_{slot}, T_{minRTT} + (i + 1)T_{slot}]$, tel que $i \in [0, N - 1]$. On suppose que le délai d'aller-retour $T_{RTT} \in [T_{minRTT}, T]$ et suit une loi de Poisson de paramètre λ . On divise $[T_{minRTT}, T]$ en des intervalles de temps de taille T_{slot} tels que :

$$P(T_{minRTT} + iT_{slot} \leq T_{RTT} < T_{minRTT} + (i + 1)T_{slot}) = \frac{\lambda^i}{i!} e^{-\lambda}, i \in [0, N - 1] \quad (4.5)$$

Par conséquent, pour la probabilité de transition q_k dans la file d'attente capteur,

$$q_k = \frac{\lambda^k}{k!} e^{-\lambda} \quad (4.6)$$

L'état $(-1, 0)$ implique qu'un nouveau paquet est prêt à être transmis dans le prochain intervalle de temps T_{slot} , c'est un état qui doit être franchi pour chaque nouvelle transmission. L'état $(-1, N - 1)$, quant à lui, veut dire que le paquet transmis ne va arriver qu'après un temps $N \times T_{slot}$.

En outre, les n boucles de commandes sont multiplexées dans le point d'accès (contrôleur). Pour cela, on ajoute un état Q à la chaîne de Markov pour modéliser l'état d'attente du paquet de commande dans la file d'attente de transmission du point d'accès avant qu'il soit en tête de cette file (figure 4.2) pour commencer la procédure DCF.

Soit R la valeur maximale de retransmission ou de l'étape de backoff. Les probabilités de transition de la matrice de transition de la chaîne de Markov de la figure 4.2 sont définies dans les équations 4.7 à 4.13. Pour une nouvelle période d'échantillonnage et après une transmission réussie des deux paquets de la boucle de commande, un nouveau paquet de mesures commence l'étape de backoff 0 et la valeur du temps de backoff est choisie uniformément dans l'intervalle $[0; W_0 - 1]$:

$$P[(0, k)|(j, 0)] = \frac{1 - p}{W_0} \quad (4.7)$$

avec $R + 1 \leq j < 2R + 1$ et $0 \leq k \leq W_0 - 1$.

A chaque fois que la procédure de backoff du paquet de commande arrive à la $2R + 1^{ieme}$ étape du backoff, l'état de l'étape de backoff est initialisé quel que soit l'état de transmission du paquet de commande (succès ou collision) :

$$P[(0, k)|(2R + 1, 0)] = \frac{1}{W_0} \quad (4.8)$$

avec $0 \leq k \leq W_0 - 1$.

Au début de chaque intervalle de temps de durée T_{slot} , le temps de backoff est décrémenté :

$$P[(j, k)|(j, k + 1)] = 1 \quad (4.9)$$

pour $0 \leq j \leq 2R + 1$ et $0 \leq k \leq W_j - 2$.

Dans le cas d'une transmission non réussie (collision) à la $j - 1^{ieme}$ étape de backoff, l'état de l'étape de backoff est incrémenté et le temps de backoff est uniformément choisi dans l'intervalle $[0, W_j - 1]$:

$$P[(j, k)|(j - 1, 0)] = \frac{p}{W_j} \quad (4.10)$$

avec $1 \leq j \leq R$ ou $R + 2 \leq j \leq 2R + 1$ et $0 \leq k \leq W_j - 1$.

Avant de commencer une nouvelle période de fonctionnement de la boucle de commande, un compte à rebours est actionné jusqu'à l'arrivée d'un nouveau paquet de mesures dans la file d'attente de transmission du système en réseau :

$$P[(-1, k)|(-1, k + 1)] = 1 \quad (4.11)$$

pour $0 \leq k \leq N - 2$.

La probabilité que la file d'attente de transmission dans le point d'accès (contrôleur) ne soit pas vide et que le paquet de commande correspondant au paquet de mesures transmis avec succès ne puisse pas commencer la procédure de backoff est définie par :

$$P[Q|Q] = q \quad (4.12)$$

La probabilité que le paquet de commande commence la procédure de backoff est définie comme suit :

$$P[(R + 1, k)|Q] = \frac{1 - q}{W_0} \quad (4.13)$$

4.4.4 Étude probabiliste du modèle markovien des réseaux IEEE 802.11 DCF

L'objectif de la modélisation stochastique Markovienne est de déterminer la probabilité de collision d'un paquet p et la probabilité p_r qu'une boucle de commande ait un paquet

pour la transmission dans un intervalle de temps T_{slot} . Cette étude est faite pour un réseau sans fil dont le mécanisme d'accès au médium est le CSMA/CA implémentant la procédure DCF. En déterminant les probabilités p et p_τ , on peut définir les expressions analytiques de T_c , S , T_{avgRTT} , G_{max} et PL_{max} . On commence tout d'abord par déterminer la distribution stationnaire de la chaîne de Markov $b_{j,k} = \lim_{t \rightarrow +\infty} Pr\{s(t) = j, b(t) = k\}$. En utilisant les lois de régularité d'une chaîne de Markov et les équations 4.3, nous obtenons :

$$\begin{aligned}
 b_{0,0} &= \frac{1}{W_0} b_{-1,0} + b_{0,1} \\
 &= \frac{1}{W_0} b_{-1,0} + \frac{1}{W_0} b_{-1,0} + b_{0,2} \\
 &= 2 \times \frac{1}{W_0} b_{-1,0} + b_{0,2} \\
 &= (W_0 - 1) \times \frac{1}{W_0} b_{-1,0} + b_{0,W_0-1} \\
 &= (W_0 - 1) \times \frac{1}{W_0} b_{-1,0} + \frac{1}{W_0} b_{-1,0} \\
 &= W_0 \times \frac{1}{W_0} b_{-1,0} \\
 &= b_{-1,0}
 \end{aligned} \tag{4.14}$$

$$\begin{aligned}
 b_{j,0} &= \frac{p}{W_j} b_{j-1,0} + b_{j,1} \\
 &= \frac{p}{W_j} b_{j-1,0} + \frac{p}{W_j} b_{j-1,0} + b_{j,2} \\
 &= 2 \times \frac{p}{W_j} b_{j-1,0} + b_{j,2} \\
 &= p b_{j-1,0} \\
 &= p^j b_{0,0}
 \end{aligned} \tag{4.15}$$

$0 \leq j \leq R$; paquet de mesures

$$\begin{aligned}
 b_Q &= (1 - p) \sum_{i=0}^R b_{i,0} + q b_Q \\
 &= (1 - p^{R+1}) b_{0,0} + q b_Q \\
 &= \frac{1 - p^{R+1}}{1 - q} b_{0,0}
 \end{aligned} \tag{4.16}$$

$$\begin{aligned}
b_{R+1,0} &= \frac{1-q}{W_0}b_Q + b_{R+1,1} \\
&= \frac{1-q}{W_0}b_Q + \frac{1-q}{W_0}b_Q + b_{R+1,2} \\
&= 2 \times \frac{1-q}{W_0}b_Q + b_{R+1,2} \\
&= (W_0 - 1) \times \frac{1-q}{W_0}b_Q + b_{R+1,W_0-1} \\
&= W_0 \times \frac{1-q}{W_0}b_Q \\
&= (1 - p^{R+1})b_{0,0}
\end{aligned} \tag{4.17}$$

$$\begin{aligned}
b_{j,0} &= p^{j-(R+1)}b_{R+1,0} \\
&= p^{j-(R+1)}(1 - p^{R+1})b_{0,0} \\
R+1 &< j \leq 2R+1; \text{ paquet de commande}
\end{aligned} \tag{4.18}$$

$$\begin{aligned}
b_{j,k} &= \frac{p}{W_j}b_{j-1,0} + b_{j,k+1} \\
&= \frac{1}{W_j}b_{j,0} + b_{j,k+1} \\
&= \frac{1}{W_j}b_{j,0} + \frac{p}{W_j}b_{j-1,0} + b_{j,k+2} \\
&= 2 \times \frac{1}{W_j}b_{j,0} + b_{j,k+2} \\
&= (W_j - 1 - k) \times \frac{1}{W_j}b_{j,0} + b_{j,W_j-1} \\
&= (W_j - 1 - k) \times \frac{1}{W_j}b_{j,0} + \frac{1}{W_j}b_{j,0} \\
&= \frac{W_j - k}{W_j}b_{j,0}
\end{aligned} \tag{4.19}$$

$$0 \leq j \leq R$$

$$1 \leq k \leq W_j - 1$$

$$b_{j,k} = \frac{W_{j-R-1} - k}{W_{j-R-1}}b_{j,0} \tag{4.20}$$

$$R+1 \leq j \leq 2R+1$$

$$1 \leq k \leq W_j - 1$$

Quand une nouvelle période est activée, un nouveau paquet de mesures est généré et doit commencer la phase du backoff. La probabilité de transition q_k dépend du délai d'aller-retour T_{RTT} de la transmission actuelle. Ce délai est composé principalement du retard, mis par chacun des deux paquets, du début de la procédure du backoff jusqu'à sa transmission réussie. De plus, un paquet doit entrer dans la procédure de backoff au moins une fois. Par conséquent, les probabilités stationnaires correspondant à chaque étape de transition dépendent du temps de transmission aller-retour actuel et sont définies par l'équation 4.21 [TCT16] :

$$\begin{aligned} b_{-1,k} &= \left(\sum_{i=k}^{N-1} \frac{\lambda^i}{i!} e^{-\lambda} \right) ((1-p) \sum_{j=R+1}^{2R} b_{j,0} + b_{2R+1,0}) \\ &= \left(\sum_{i=k}^{N-1} \frac{\lambda^i}{i!} e^{-\lambda} \right) (1-p^{R+1}) b_{0,0} \end{aligned} \quad (4.21)$$

Appliquons une condition de normalisation, la relation suivante de l'équation 4.22 est obtenue à partir de l'équation 4.3 :

$$\sum_{j=0}^R \sum_{k=0}^{W_j-1} b_{j,k} + \sum_{j=R+1}^{2R+1} \sum_{k=0}^{W_j-R-1-1} b_{j,k} + \sum_{k=0}^{N-1} b_{-1,k} + b_Q = 1 \quad (4.22)$$

On utilise les équations de 4.14 à 4.21 pour faire apparaître le $b_{0,0}$ dans les 4 termes de l'équation 4.22. On obtient ainsi l'expression analytique de la probabilité $b_{0,0}$:

$$\begin{aligned} b_{0,0} &= \left[\frac{2-p^{R+1}}{2} \left(\frac{1-p^{R+1}}{1-p} - \sum_{j=0}^R W_j p^j \right) + \sum_{k=0}^{N-1} \sum_{i=k}^{N-1} (1-p^{R+1}) \frac{\lambda^i}{i!} e^{-\lambda} \right. \\ &\quad \left. + \frac{1-p^{R+1}}{1-q} \right]^{-1} \end{aligned} \quad (4.23)$$

or la taille de la fenêtre W_j double à chaque passage d'une étape de backoff à une autre et par conséquent $W_j = 2^j W_0$. L'équation 4.23 devient alors :

$$\begin{aligned} b_{0,0} &= \left[\frac{2-p^{R+1}}{2} \left(\frac{1-p^{R+1}}{1-p} - W_0 \frac{1-(2p)^{R+1}}{1-2p} \right) + \sum_{k=0}^{N-1} \sum_{i=k}^{N-1} (1-p^{R+1}) \frac{\lambda^i}{i!} e^{-\lambda} \right. \\ &\quad \left. + \frac{1-p^{R+1}}{1-q} \right]^{-1} \end{aligned} \quad (4.24)$$

La probabilité q , qui représente la probabilité que la file d'attente de transmission dans le point d'accès ne soit pas vide, correspond à la probabilité qu'un paquet soit déjà présent dans la file d'attente de transmission du point d'accès [GFS12]. Dans ce cas, le paquet de commande correspondant au paquet de mesures transmis ne peut pas commencer la

procédure de backoff DCF vu qu'il n'est pas en tête de la file d'attente du point d'accès. On définit alors la probabilité $1 - q$ comme étant la probabilité que le paquet de commande soit en tête de la file d'attente et prêt à commencer la procédure DCF. La probabilité q est exprimée dans l'équation 4.25 en supposant que le taux d'arrivée de paquets dans la file d'attente de transmission du contrôleur (point d'accès) suit une loi de Poisson de paramètre $n\lambda$. En effet, les n boucles de commandes ont un taux d'arrivée moyen de paquets dans leurs files d'attente λ , que nous avons défini dans la loi de probabilité de T_{RTT} , et ils ont la même période d'échantillonnage et les mêmes tailles de paquets.

$$q = P(nbr > 0) = 1 - P(nbr = 0) \quad (4.25)$$

avec nbr le nombre de paquets dans la file d'attente du contrôleur.

Le travail de [Szt12] a détaillé des lois pour la théorie des files d'attente en particulier la théorie de $M/M/1/K$. En effet, dans un système de file d'attente $M/M/1/K$, les paquets arrivent dans une file d'attente de capacité K au niveau d'un seul serveur de lois de commande (le contrôleur) selon une loi de Poisson. On définit également un temps de service pour chaque paquet comme étant le temps écoulé entre l'instant où le paquet entre dans la procédure du backoff et l'instant où il est transmis avec succès ou bien l'instant où la transmission a échoué. Ce temps de service suit une loi exponentielle de paramètre μ et le temps moyen de service est ainsi égal à $\frac{1}{\mu}$. [Szt12] a établi l'expression de $P(nbr = 0)$ suivant la théorie de la file d'attente $M/M/1/K$. La probabilité q est alors définie par l'équation 4.26 :

$$q = 1 - \frac{1 - \rho}{1 - \rho^{K+1}}; \rho \neq 1 \quad (4.26)$$

Le paramètre ρ représente l'intensité du trafic au niveau du point d'accès, c'est le produit entre le taux d'arrivée de paquets et le temps moyen de service d'un paquet. ρ est alors égal à $\frac{n\lambda}{\mu}$.

On étudie la performance du trafic aller-retour de chaque boucle de commande. Pour cela, soit p_τ la probabilité qu'une boucle de commande transmette dans une étape de backoff dans un intervalle de temps choisi aléatoirement. En effet, une boucle de commande transmet quand le temps de backoff au niveau du capteur ou du contrôleur arrive à 0. La probabilité p_τ est définie par l'équation 4.27 suivante :

$$p_\tau = \sum_{j=0}^{2R+1} b_{j,0} = (2 - p^{R+1}) \frac{1 - p^{R+1}}{1 - p} b_{0,0} \quad (4.27)$$

On peut maintenant définir l'expression de la probabilité de collision p : c'est la probabilité qu'au moins un des $n - 1$ nœuds transmette dans un intervalle de temps de durée T_{slot} . La probabilité p est alors déterminée par l'équation 4.28 comme suivante :

$$p = 1 - (1 - p_\tau)^{n-1} \quad (4.28)$$

On peut aussi, à ce stade, déterminer les expressions analytiques de la probabilité d'une transmission réussie p_s et la probabilité p_b que le canal de communication soit occupé pendant l'intervalle de temps de durée T_{slot} . Ces expressions sont définies par les équations 4.29 et 4.30 dans [Bia00] :

$$p_s = np_\tau(1 - p_\tau)^{n-1} \quad (4.29)$$

$$p_b = 1 - (1 - p_\tau)^n \quad (4.30)$$

Le modèle Markovien développé a servi d'outil pour établir les expressions analytiques de p , p_τ , p_s et p_b . Cette étude analytique va permettre de déterminer mathématiquement la valeur critique de la période d'échantillonnage T_c qui garantit une bonne QoS et par conséquent une bonne QoC en se servant des équations établies. C'est une condition nécessaire mais pas suffisante. Elle va également servir à déterminer le débit de saturation, le délai moyen de transmission aller-retour, la valeur tolérable de la gigue et du taux de pertes de paquets dans le réseau.

4.5 Définition des paramètres de la QoS

4.5.1 Débit de saturation et période d'échantillonnage critique

TABLE 4.1 – Notations

Notation	Explications
$E[P]$	taille moyenne d'un paquet
$E[slot]$	durée moyenne de l'intervalle de temps (time slot)
T_H	durée de la transmission de l'en-tête d'un paquet
$T_{E[P]}$	durée de la transmission de la charge utile moyenne
T_{SIFS}	délai SIFS
T_{Ack}	durée de transmission d'un acquittement ACK
T_{DIFS}	délai DIFS
$T_{acktimeout}$	délai d'attente d'un acquittement ACK

Le débit de saturation S est le débit maximal atteint en respectant les contraintes temps réel critiques. Il est défini comme étant la mesure de la quantité d'informations transmises par seconde avec succès. On note T_s la durée moyenne pendant laquelle le canal de communication est occupé par une transmission réussie et T_{col} la durée pendant laquelle le canal est occupé à cause d'une collision. Les durées T_s et T_{col} sont définies par les équations 4.31 et 4.32 :

$$T_s = T_H + T_{E[P]} + T_{SIFS} + T_{Ack} + T_{DIFS} \quad (4.31)$$

$$T_{col} = T_H + T_{E[P]} + T_{DIFS} + T_{acktimeout} \quad (4.32)$$

avec $E[P]$, T_{SIFS} , T_{Ack} , T_{DIFS} , T_H et $T_{acktimeout}$ définis dans la table 4.1. Le débit de saturation S est alors défini dans l'équation 4.33 suivante :

$$\begin{aligned} S &= \frac{p_s E[p]}{E[slot]} \\ &= \frac{p_s E[p]}{(1 - p_b)T_{slot} + p_s T_s + (p_b - p_s)T_{col}} \end{aligned} \quad (4.33)$$

Ce terme $p_s E[p]$ représente la quantité moyenne de la charge utile transmise avec succès dans un intervalle de temps et le terme $p_b - p_s$ définit la probabilité que le canal de communication est occupée à cause d'une collision.

Les boucles de commande ont toutes la même période d'échantillonnage. La période d'échantillonnage critique temps réel T_c qui garantit la satisfaction d'un bon niveau de la QoS est déterminée analytiquement par l'équation 4.34 :

$$T_c = \frac{2nE[P]}{S} + \bar{w} \quad (4.34)$$

où \bar{w} représente la durée d'attente moyenne dans la file d'attente du point d'accès.

Pour résoudre le modèle Markovien proposé, notre étude est basée sur la théorie de la file d'attente $M/M/1/K$ en supposant que le taux d'arrivée de paquets suit une loi de Poisson de paramètre $n\lambda$ et que le temps de service d'un paquet suit une loi exponentielle de paramètre μ . Le temps moyen d'attente \bar{w} dans la file d'attente (de capacité K) du point d'accès est défini dans l'équation 4.35 suivante :

$$\bar{w} = \frac{\rho + \rho^{K+1}(\rho K - K - 1)}{n\lambda(1 - \rho)(1 - \rho^K)} \quad (4.35)$$

4.5.2 Retard moyen aller-retour

Le délai de transmission d'un paquet est le temps écoulé entre l'instant où le paquet est mis en service (paquet placé en tête de la file de la couche MAC) et l'instant où le paquet est bien reçu au nœud destination. En se basant sur l'analyse faite dans [CBV03] et la théorie des files d'attente, le temps moyen aller-retour T_{avgRTT} pour la transmission des paquets de mesures et de commande est défini par l'équation 4.36 :

$$T_{avgRTT} = 2E[X] * E[slot] + \bar{w} \quad (4.36)$$

Le terme $E[X]$, défini dans l'équation 4.37 [CBV03], représente le nombre moyen d'intervalles de temps de durée T_{slot} nécessaires pour une transmission réussie d'un paquet.

$$\begin{aligned} E[X] &= \sum_{j=0}^R \frac{(p^j - p^{R+1}) W_j + 1}{1 - p^{R+1}} \frac{1}{2} \\ &= \frac{W_0(1 - (2p)^{R+1})(1 - p) + (1 - 2p)(1 - p^{R+1})}{2(1 - 2p)(1 - p)(1 - p^{R+1})} \\ &\quad - \frac{(p)^{R+1}(1 - 2p)(1 - p)(W_0(2^{R+1} - 1) + R + 1)}{2(1 - 2p)(1 - p)(1 - p^{R+1})} \end{aligned} \quad (4.37)$$

avec $W_j = 2^j W_0$.

Le ratio $\frac{p^j - p^{R+1}}{1 - p^{R+1}}$ représente la probabilité que le paquet arrive à la $j^{\text{ième}}$ étape du backoff et $\frac{W_j + 1}{2}$ définit la valeur moyenne du temps de backoff dans la $j^{\text{ième}}$ étape du backoff. Le délai moyen aller-retour T_{avgRTT} n'est pas déterminé seulement avec l'équation 4.36. Comme T_{RTT} suit une loi de Poisson de paramètre λ sur $[T_{minRTT}, T]$ où l'intervalle $[T_{minRTT}, T]$ est divisé en des intervalles de temps T_{slot} , sa valeur moyenne T_{avgRTT} est définie par l'équation 4.38 :

$$T_{avgRTT} = T_{minRTT} + \lambda T_{slot} \quad (4.38)$$

avec T_{minRTT} la durée minimale de temps de transmission aller-retour qui est égale à $2 \times T_s$ et λ la moyenne pour une loi de Poisson.

Pour déterminer l'intensité du trafic ρ , il faut chercher tout d'abord à calculer $\frac{1}{\mu}$ qui représente le temps moyen de service d'un paquet. Il est défini comme suit dans l'équation 4.39 :

$$\frac{1}{\mu} = E[X] * E[T_{slot}] \quad (4.39)$$

4.5.3 Taux de pertes de paquets

Le taux de pertes de paquets PL dû aux collisions est défini par l'équation 4.40 :

$$PL = 100 \times p^{R+1} \quad (4.40)$$

avec p^{R+1} représente la probabilité de perte d'un paquet à cause d'une collision. Et comme la probabilité de collision $p \in [0, 1]$, l'augmentation du nombre limite de retransmissions R réduit le taux de pertes de paquets dans le réseau.

4.5.4 Gigue

Dans le cas d'un WNCS, la gigue J est la variation du délai de transmission à différents instants d'échantillonnage. La valeur de la gigue est la différence entre la durée actuelle de

transmission aller-retour T_{RTT} et sa valeur moyenne T_{avgRTT} . En se basant sur la contrainte temps réel de l'inégalité 2.1, la valeur maximale de la gigue, dans les conditions critiques, est définie comme suit :

$$J_{max} = T_c - T_{avgRTT} \quad (4.41)$$

Cette valeur, définie par les performances du réseau, va être validée dans le prochain chapitre en étudiant les performances de la boucle de commande en terme de temps de réponse.

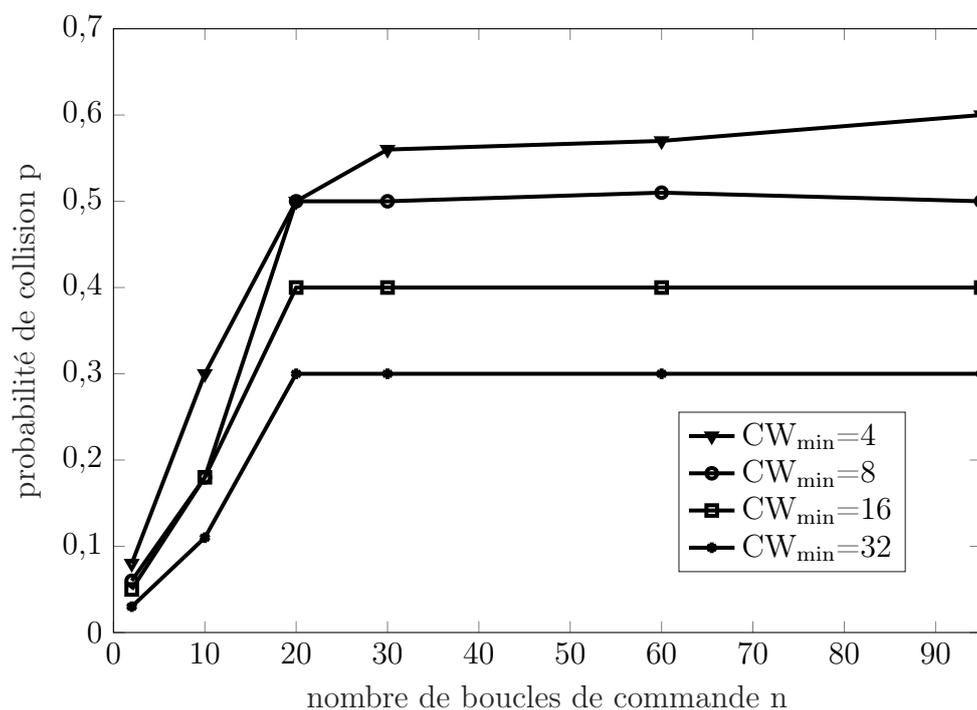
4.6 Cas d'études

On considère un réseau sans fil 802.11b composé de n boucles de commande : n robots dans la zone de couverture d'un point d'accès pour assurer la connectivité. Le choix du type de réseau sans fil vient du fait que les robots de test dont nous disposons sont équipés de cartes réseau 802.11b. Les paramètres du réseau sont définis dans le tableau 4.2. Nous

TABLE 4.2 – Les paramètres du réseau

taille d'un paquet	130 octets
taille de l'en-tête MAC	272 bits
taille de l'en-tête dans la couche physique PHY	192 bits
ACK	112 bits + PHY
débit	11Mbit/s
T_{slot}	20 μ s
T_{SIFS}	10 μ s
T_{DIFS}	50 μ s
R	7
K	2

allons analyser par la suite les valeurs de p , S , T_c and T_{avgRTT} en faisant varier la taille minimale de la fenêtre de contention CW_{min} , qui correspond à W_0 dans notre analyse, telle que $CW_{min} \in \{4, 8, 16, 32\}$. Nous avons utilisé la fonction Matlab *fsolve* pour résoudre le système d'équations non linéaires composé de 7 équations définies à partir des équations 4.24, 4.26, 4.27, 4.28, 4.36, 4.38, 4.39. En effet, la taille minimale de la fenêtre de contention CW_{min} impacte la probabilité de collision. Quand CW_{min} est faible, le nombre de tentatives de transmission augmente ainsi que le taux de collisions et plus particulièrement pour un grand nombre de boucles de commande dans le réseau comme le montre la figure 4.3.

FIGURE 4.3 – Probabilité de collision pour différentes valeurs de CW_{min}

Par conséquent, le débit de saturation S associé à la plus petite valeur de CW_{min} est le plus pénalisé et la période T_c est la plus grande comme le montrent les figures 4.4 et 4.5. Pour le retard moyen T_{avgRTT} , la figure 4.6 montre qu'il augmente quand CW_{min} augmente parce que la taille de la fenêtre de contention a un impact sur la durée d'attente du paquet (lors de la procédure de backoff) dans le nœud émetteur.

Le débit de saturation S , déterminé théoriquement grâce à notre modèle analytique, dépend du nombre de boucles de commande qui partagent le réseau sans fil et de la taille initiale de la fenêtre de contention. En effet, si la taille de la fenêtre de contention CW augmente dans le cas d'un grand nombre n de boucles de commande, le débit S est performant mais il est pénalisé quand n devient petit. Une grande valeur de CW engendre un débit limité pour un petit nombre n de boucles de commande. Le nœud émetteur attend pendant une durée considérable avant de tenter d'accéder au canal de communication. Quand le nombre de boucles de commande n augmente, le débit se dégrade.

Par ailleurs, la période d'échantillonnage critique T_c augmente si n augmente afin d'éviter la dégradation de la QoS. Si la période d'échantillonnage T est réduite au-dessous de T_c pour chaque boucle de commande, les exigences de la QoS ne sont pas satisfaites et provoquent une surcharge dans le réseau et de longs délais. Dès lors, la performance du système se dégrade et une bonne QoS n'est plus garantie. En outre, les figures 4.6 et 4.7 montrent que le délai T_{avgRTT} et la gigue (paramètre important des applications temps réel) augmentent avec l'augmentation du nombre n de boucles de commande. La valeur maximale de la gigue

est liée au retard moyen d'aller-retour et à la période d'échantillonnage (voir équation 4.41). Quand le nombre n augmente, la gigue augmente également. Tout cela montre que l'augmentation du nombre n de boucles de commande détériore la performance du réseau. En outre, le temps d'attente dans la file du contrôleur croît en fonction du nombre n . Il impacte les valeurs de T_c , T_{avgRTT} et de la gigue. Cela explique l'évolution similaire de ces trois derniers paramètres pour les différentes valeurs de CW_{min} .

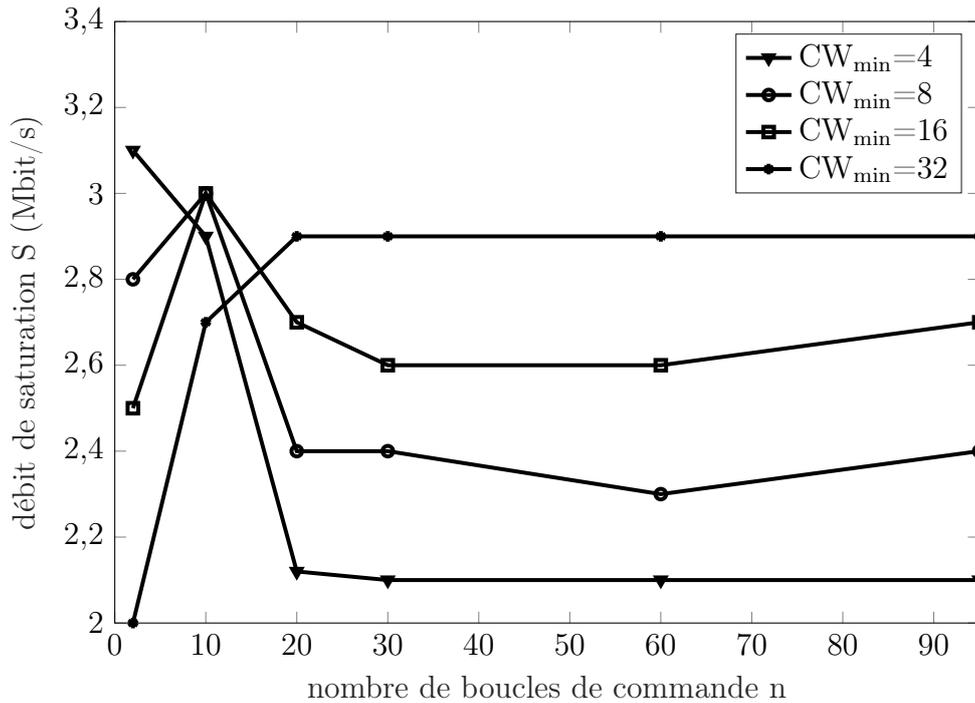


FIGURE 4.4 – Débit de saturation pour différentes valeurs de CW_{min} .

4.7 Conclusion

Le comportement stochastique des réseaux sans fil introduit des incertitudes qui peuvent provoquer une dégradation. Ces incertitudes et leurs influences sur la QoS vont être modélisées par un RB. Dans le RB qui va être présenté dans la suite, les limites de la bonne performance doivent être définies pour obtenir des résultats pertinents. Étudier la dynamique des réseaux sans fil est une tâche primordiale pour définir les performances de la QoS dans un SCR sans fil. Un modèle markovien a alors été établi pour caractériser le trafic périodique de chaque boucle de commande. Ce modèle a servi pour définir les expressions mathématiques des paramètres de la QoS tels que le débit de saturation et le retard moyen d'aller-retour à partir de la distribution stationnaire de la chaîne de Markov. La période d'échantillonnage critique au-dessous de laquelle les performances du réseau se dégradent

a aussi été définie.

Cependant, l'analyse du réseau seule n'est pas suffisante pour étudier les performances d'un SCR sans fil. Les performances du contrôleur doivent également être prises en considération.

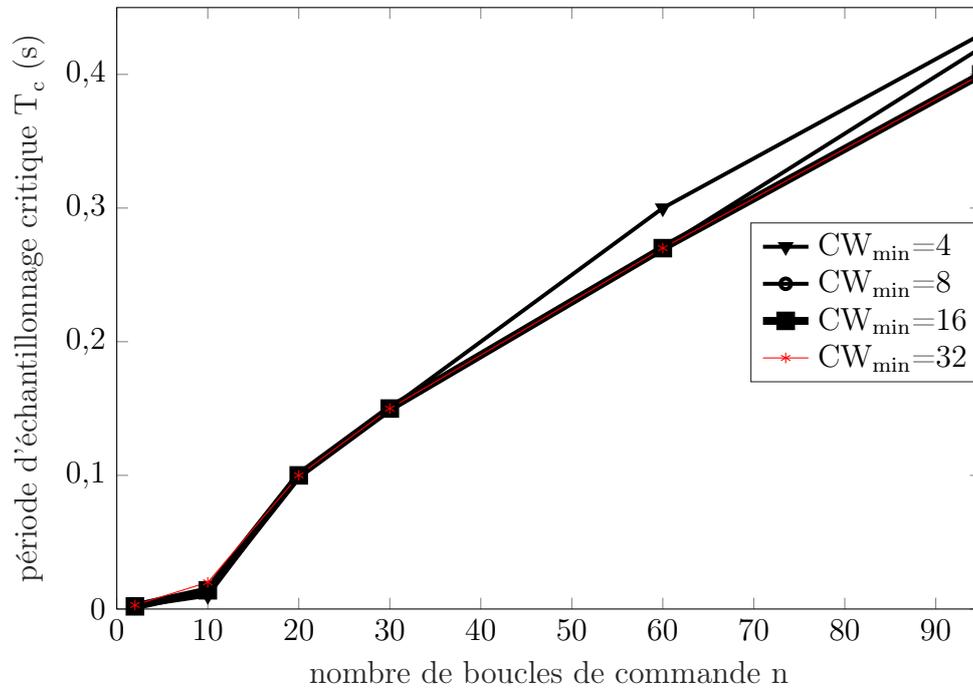
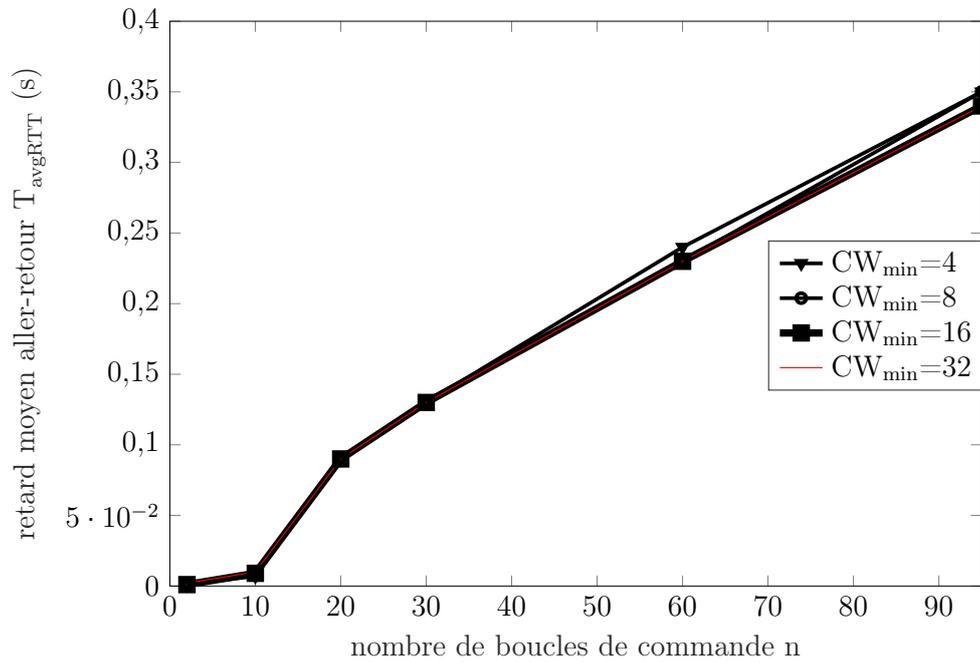
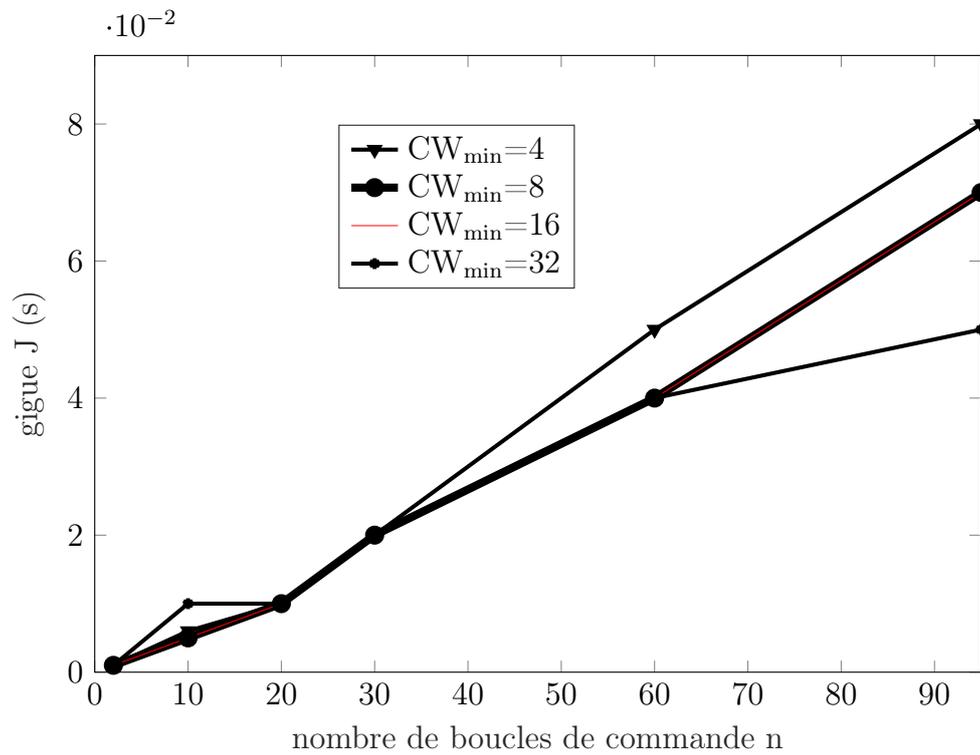


FIGURE 4.5 – Période d'échantillonnage critique pour différentes valeurs de CW_{min}

FIGURE 4.6 – Délai moyen d’aller-retour pour différentes valeurs de CW_{min} FIGURE 4.7 – Gigue pour différentes valeurs de CW_{min}

Réseau Bayésien pour le diagnostic d'un robot mobile commandé en réseau

Sommaire

5.1	Introduction	77
5.2	Définition du système	78
5.2.1	Architecture du système	78
5.2.2	Étude de l'effet de la dégradation de la QoS sur la QoC	81
5.3	Réseau bayésien pour le diagnostic : étude qualitative	85
5.3.1	Module bayésien de la partie commande	89
5.3.2	Module bayésien de la partie opérative	91
5.3.3	Module bayésien de la partie communication	93
5.4	Réseau bayésien pour le diagnostic : étude quantitative	98
5.4.1	Partie commande	98
5.4.2	Partie opérative	100
5.4.3	Partie communication	102
5.5	Scénarios de diagnostic et d'aide à la décision	106
5.6	Reconfiguration de l'architecture de diagnostic	112
5.6.1	Commutation du diagnostic distribué au diagnostic embarqué	112
5.6.2	Commutation du diagnostic embarqué au diagnostic distribué	116
5.6.3	Scénario de diagnostic et de commutation	119
5.7	Conclusion	124

5.1 Introduction

Dans un cadre de co-design, l'adaptation de l'architecture de commande ou du réseau de communication est une solution pour améliorer le comportement du système. Une méthode de détection de la dégradation de la performance du système et de prise de décision

est primordiale. Dans ce chapitre, nous présentons une méthode de diagnostic basée sur les RBs distribués modulaires pour le diagnostic d'un SCR sans fil. L'intérêt de l'utilisation des RBs modulaires a été détaillé dans le chapitre 3. La dépendance causale entre la QoS et la QoC du système est une relation probabiliste qui justifie l'utilisation de RB. Cette dépendance entre la QoS et la QoC quantifie la satisfaction, ou non, des exigences de l'application. Par analogie avec la qualité d'expérience (QoE), la dépendance entre la QoS et la QoC évalue la qualité de service fourni. La QoE désigne la qualité de service perçue par un utilisateur final, c'est une évaluation, qui peut être soit subjective soit objective, du service fourni qui sert à améliorer la QoS du réseau [RMM11]. L'évaluation de la QoS à partir des observations de la QoC, en utilisant le RB, est une évaluation objective des performances du réseau d'un point de vue commande. Dans le cadre de nos travaux, le RB sert d'une part à prendre des décisions et d'autre part à effectuer un diagnostic en ligne. Il permet de déclencher la procédure de commutation de la commande débarquée à la commande embarquée ou vice-versa ou bien de commuter sur un nouveau réseau de communication (ou station de contrôle). La QoS est modélisée et quantifiée à l'aide de ses paramètres : débit, délai, gigue, perte de paquets, connectivité. L'environnement et les conditions de fonctionnement sont pris en compte pour montrer leurs influences sur la QoS, la QoC et même l'état du robot. Il existe des conditions de fonctionnement qui ajoutent de l'indéterminisme sur le comportement du système et d'autres qui ont une influence déterministe. Le comportement déterministe a été défini dans le chapitre 4 où nous avons détaillé les conditions pour avoir une bonne QoS.

Nous présentons dans ce chapitre une méthode de diagnostic dynamique et des procédures de commutation d'un RB distribué à un RB monolithique et vice-versa. Nous avons développé une stratégie de commutation en nous basant sur les résultats du RB pour le diagnostic du robot mobile commandé en réseau sans fil depuis une station distante que nous allons détailler dans la section 5.5.

5.2 Définition du système

5.2.1 Architecture du système

Un robot mobile commandé par une station distante est un SCR sans fil où un contrôleur débarqué est implémenté sur la station et où l'ensemble des actionneurs et des capteurs est implanté sur le robot comme le montre la figure 5.1. La mission principale du robot est l'atteinte d'une cible, dont les coordonnées (X_{ref}, Y_{ref}) sont données par l'utilisateur, en évitant des obstacles. Deux boucles de commande en cascade ont été implémentées. Une première boucle de commande est implémentée via le réseau sans fil pour la régulation de la position du robot en calculant les vitesses linéaire v et angulaire ω (voir figure 5.1) telles

que v et ω sont les deux lois de commande non linéaires proposées par [Toi+11], [TCK07].

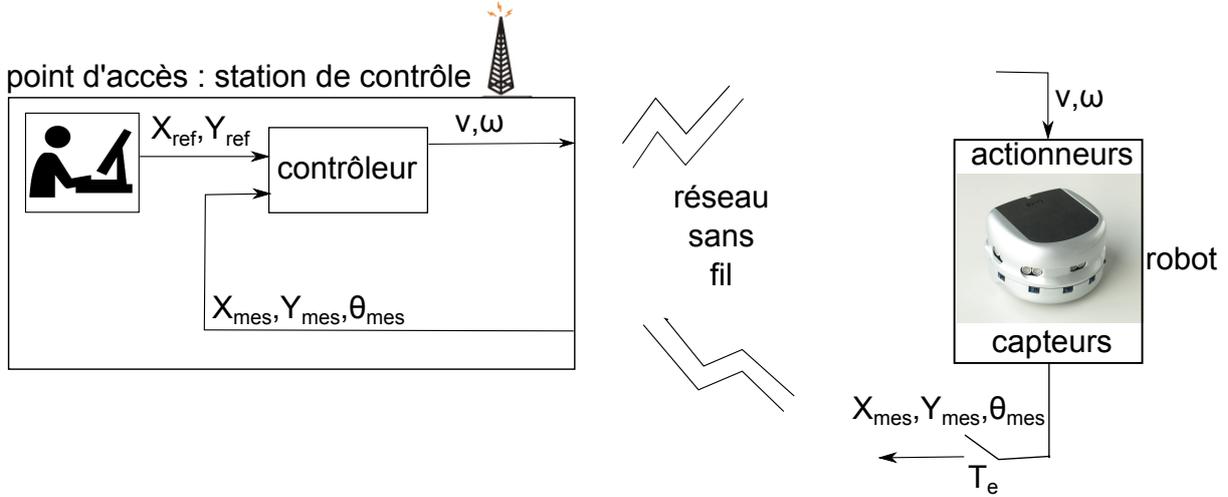


FIGURE 5.1 – Définition du système

La deuxième boucle de commande est implémentée localement sur le robot pour la régulation des deux vitesses angulaires ω_r et ω_l des moteurs gauche et droit où nous utilisons un régulateur PI comme le montre la figure 5.2. Le premier contrôleur envoie via le réseau les valeurs de la commande sous la forme de vitesses linéaire et angulaire au robot après avoir reçu les mesures des capteurs la position et l'orientation du robot ($X_{mes}, Y_{mes}, \theta_{mes}$) à chaque période d'échantillonnage T_e . La station de contrôle est le serveur des lois de commande et des scénarios de prises de décision pour le robot. Le robot calcule localement la nouvelle valeur de la consigne en vitesses angulaires quand il reçoit les nouvelles valeurs de v et ω . Le contrôleur local reçoit alors les nouvelles consignes en vitesses angulaires ω_{ref_k} et déclenche l'exécution de la boucle de commande locale. Dans le but de réduire les calculs et d'optimiser la consommation d'énergie au niveau du robot, le contrôleur local est un algorithme de contrôle simple qui est un régulateur PI qui calcule la commande en tension u_k appliquée à chaque moteur (voir figure 5.2). Le régulateur PI a été configuré de manière à avoir comme performance un temps de réponse de $100ms$ [Mec10] (voir annexe B). Dans le cas du régulateur PI discret, la période d'échantillonnage est choisie selon l'inégalité 2.2. Soit $T'_e = 20ms$ la période d'échantillonnage de la boucle locale [Mec10]. La boucle de commande en réseau, fournisseur du signal de référence à la boucle locale, doit avoir une période d'échantillonnage T_e plus grande que T'_e de telle sorte que la boucle locale ait une dynamique plus rapide [LZ05] :

$$5 \times T'_e \leq T_e \quad (5.1)$$

Si la période d'échantillonnage T_e est très grande, le robot ne sera pas réactif et n'arrivera pas à corriger proprement les erreurs. Par contre, si elle est très petite, les mesures ne

seront pas significatives et le robot, qui a une capacité de calcul réduite, va être surchargé par des calculs inutiles. Dans une approche de co-design, la performance du réseau doit être prise en considération. En effet, l'étude analytique faite dans le chapitre 4 permet de déterminer la plus petite période d'échantillonnage (la période d'échantillonnage critique) T_c que le réseau peut supporter. Elle est telle que :

$$T_c \leq T_e \quad (5.2)$$

Cette période d'échantillonnage est définie en fonction du nombre de boucles de commande qui partagent le réseau comme nous l'avons montré dans l'équation 4.34. Pour pouvoir choisir la bonne période d'échantillonnage T_e , il faut prendre en considération, les deux équations 5.1 et 5.2. Pour le cas d'un seul robot, i.e une seule boucle de commande, nous avons choisi T_e égale à $400ms$, ainsi $T_e = 20 \times T'_e$ et $T_e \geq 3 \times 10^{-3}s$. La valeur critique de la période d'échantillonnage $T_c = 3 \times 10^{-3}s$ est définie à l'aide de l'équation 4.34 du chapitre 4 avec le nombre de boucles de commande n égal à 1.

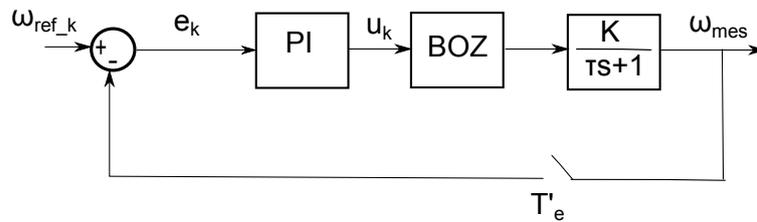


FIGURE 5.2 – Boucle de commande locale

Le réseau de communication sans fil intégré dans la boucle de commande entre le contrôleur et les actionneurs d'une part et entre les capteurs et le contrôleur d'autre part (voir figure 5.1) est un réseau IEEE 802.11b. La période d'échantillonnage critique T_c a été déterminée selon les paramètres et les caractéristiques de ce réseau qui sont définis dans la table 4.2. Les données capteurs et contrôleur sont cruciales pour garantir la stabilité du système. L'utilisation du réseau sans fil introduit des retards et des pertes de paquets qui dégradent la performance et le déroulement de la mission du robot. La retransmission est acceptable jusqu'à un nombre maximal de tentatives de transmissions R (défini dans la table 4.2) qui garantit le respect du délai et des instants limites de réception (avant le début d'une nouvelle période d'échantillonnage).

En outre, le robot ne peut communiquer avec la station de contrôle que s'il est dans sa zone de couverture, comme dans le cas du robot connecté à la station qui évolue dans la zone de couverture 3 (figure 1.1). Dans ce cas, le contrôleur débarqué est le calculateur des lois de commandes. Si le robot s'éloigne de cette zone, il perd la communication et commute sur son contrôleur embarqué (fonctionnement autonome) pour garantir la continuité de sa mission jusqu'à ce qu'il trouve une nouvelle station de contrôle (un nouveau contrôleur distant). Le mode de communication dans notre application est un mode infrastructure où

la station de contrôle est considérée comme étant le point d'accès gestionnaire de l'accès au réseau, et elle communique avec les autres stations de contrôle via un réseau filaire.

L'intégration d'un réseau sans fil dans la boucle de commande génère des problèmes qui dégradent les performances du système. Plusieurs paramètres, qui représentent la QoS du réseau, ont un impact sur la performance du système commandé en réseau et sur sa QoC. Il faut donc établir une stratégie de diagnostic qui permet de déterminer la cause de la défaillance. Notre stratégie de diagnostic doit représenter les dépendances probabilistes entre les défauts du réseau et leur impact sur la QoC. Si le réseau de communication ne peut plus garantir la QoS requise, la station envoie un ordre de rupture de connexion au robot. Par conséquent, le robot est face à deux solutions possibles : soit il commute sur son contrôleur embarqué, soit il commute sur un nouveau contrôleur débarqué comme le montre le scénario de la zone 1 dans la figure 1.1. Toutes ces prises de décisions de l'approche de co-design doivent être bien établies.

5.2.2 Étude de l'effet de la dégradation de la QoS sur la QoC

Pour mieux comprendre la dynamique du robot commandé en réseau et valider la co-conception faite, un outil Matlab de co-simulation réseau-commande *Truetime* développé par [CHO10] est utilisé. La co-simulation consiste à faire fonctionner la commande et le réseau en même temps pour avoir des résultats plus précis et fidèles à la dynamique du système réel. Elle permet d'étudier l'interaction entre deux domaines différents et de déterminer l'impact du réseau sur la stratégie de commande, et vice-versa, plus particulièrement pour les réseaux sans fil qui sont non fiables et non déterministes [CHO10], [Fer+13]. Pour analyser le comportement du système en fonction de la dynamique du réseau, la mission du robot simulée correspond à un acheminement vers une position cible en évitant les obstacles. Il commence depuis une position initiale $(X_0, Y_0) = (0, 0)$ avec une orientation initiale $\theta_0 = 0$. Dans un premier scénario de simulation, où il n'y a pas de problème de dégradation de la QoC, le robot évolue dans la zone de couverture de la station de contrôle de coordonnées $(X_{station}, Y_{station}) = (1, 1)$ qui est estimée à 100 mètres dans un réseau 802.11b à l'intérieur pour une puissance de signal d'émission de $13dBm$ et un RSSI de $-80dBm$. Le résultat de simulation de la figure 5.3 montre le comportement de référence quand la QoC est bonne.

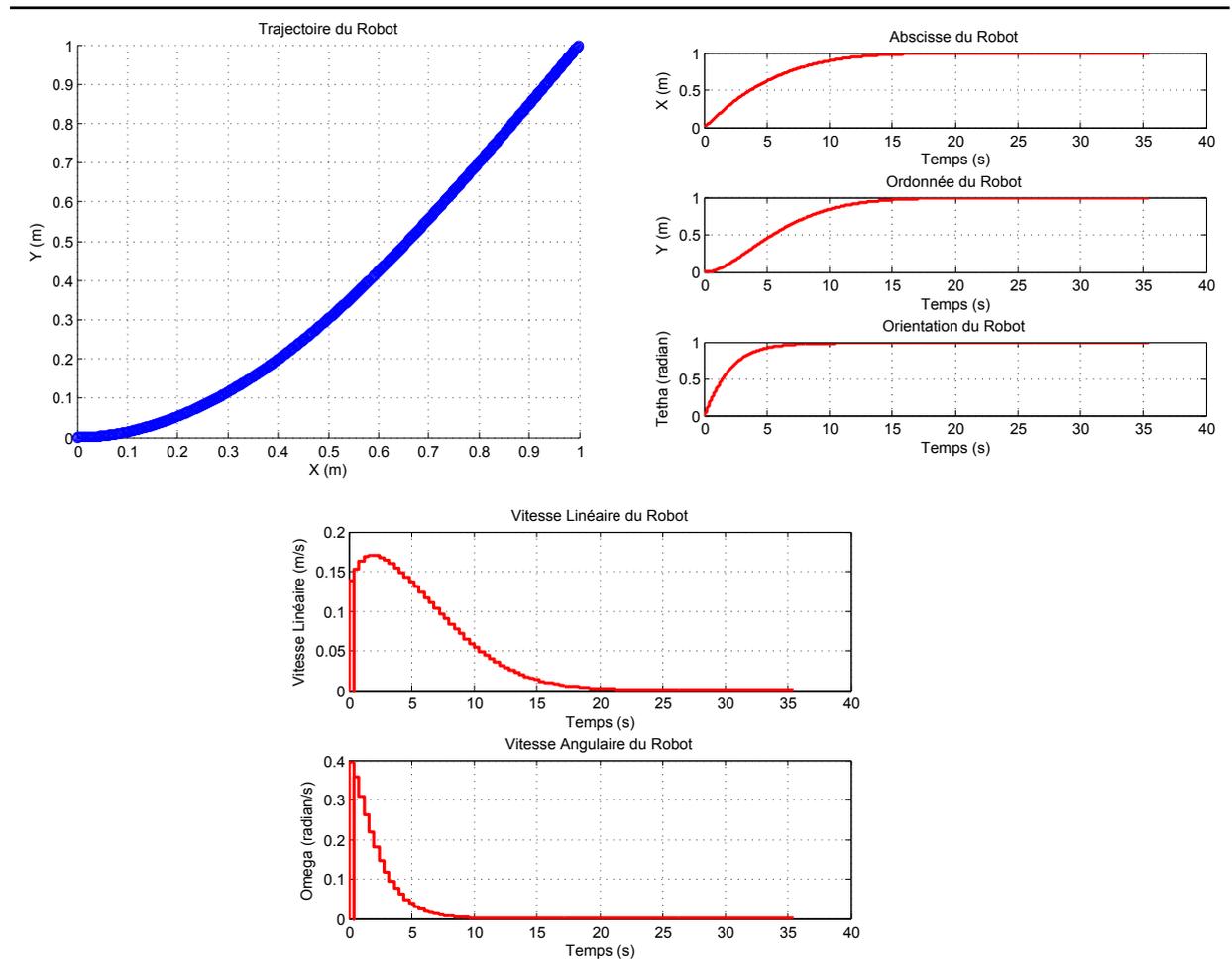


FIGURE 5.3 – Comportement du robot en simulation

Le robot atteint sa position cible de coordonnées $(X_{cible}, Y_{cible}) = (1, 1)$ et se stabilise après 20s comme le montre la figure 5.3. En guise de comparaison et de validation du système simulé, la trajectoire simulée et la trajectoire suivie par le robot lors de l'expérimentation sont présentées sur les figures 5.4. La dynamique simulée est proche de la dynamique du robot Khepera 3 utilisé dans l'expérimentation comme le montrent les figures 5.4.

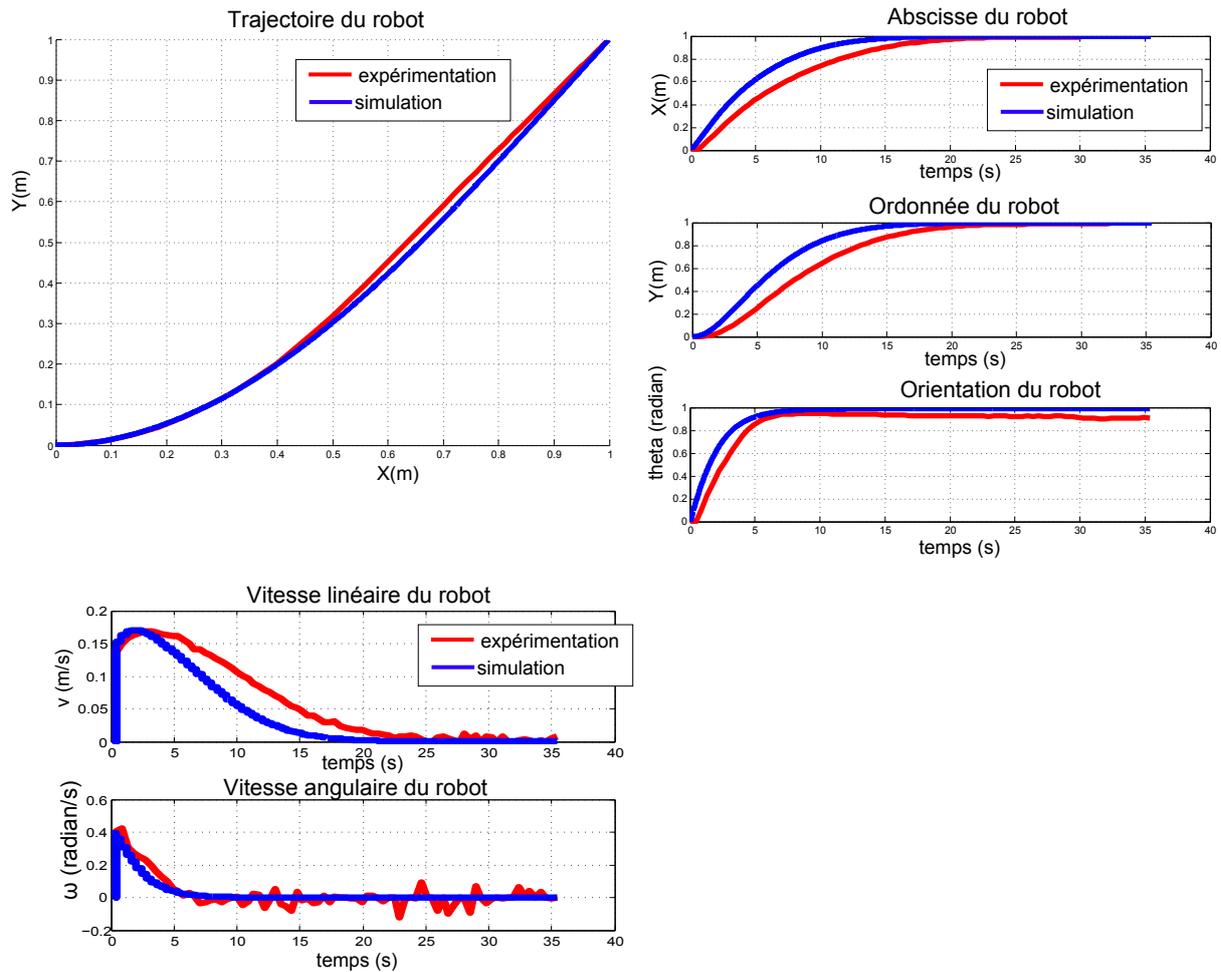


FIGURE 5.4 – Comparaison entre les comportements du robot en simulation et en expérimentation

La différence est due aux effets du frottement et aux erreurs de mesures au niveau des capteurs des deux moteurs. Ces erreurs de mesures au niveau des capteurs des moteurs apparaissent dans la courbe de la vitesse angulaire ω en fonction du temps de la figure 5.4. Le robot cherche à corriger sa position par rapport à la position cible, d'où l'apparition de quelques oscillations à la fin de sa trajectoire.

Pour la mission d'évitement d'obstacles, les capteurs d'obstacles doivent être capables de détecter l'obstacle à partir d'une distance de sécurité définie dans notre cas entre le centre de masse du robot et l'obstacle : $R_{robot} + distance_{détection} = 6.5 + 8 = 13.5cm$ où R_{robot} est le rayon du robot et $distance_{détection}$ est la distance du début de l'évitement d'obstacle. En effet, les capteurs infra-rouges du robot (placés sur le pourtour du robot) commencent à détecter les obstacles à partir 12cm de distance, l'évitement d'obstacle commence quand il est à 8cm de l'obstacle. Le robot évite l'obstacle en toute sécurité comme le montre la figure

5.5. Les réseaux sans fil ont des dynamiques stochastiques qui influencent le comportement

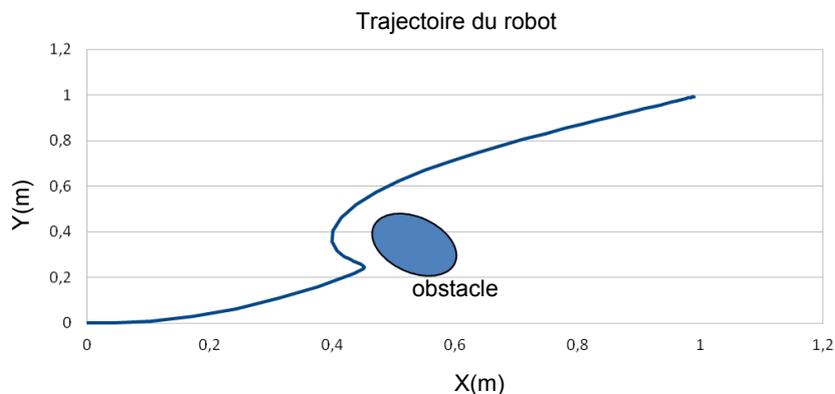


FIGURE 5.5 – Évitement d'obstacles

des systèmes dont l'architecture est distribuée. L'effet de la dégradation de la QoS sur la performance d'un SCR sans fil, c'est-à-dire la QoC du robot est l'objectif de cette étude. Le robot doit envoyer chaque $400ms$ des données sur sa position et son orientation au contrôleur qui calcule les nouvelles valeurs de commande et les envoie aux actionneurs du robot. Si les actionneurs ne reçoivent pas les nouvelles commandes en vitesses v et ω , la mission du robot risque d'être perturbée. Ils exécutent les anciennes valeurs de consignes au lieu de celles qui pourraient lui permettre d'évoluer et de corriger son comportement vers sa destination si cela était nécessaire. Dans un premier scénario, l'impact du taux de pertes de paquets sur la mission du robot est illustré sur la figure 5.6. Pour un taux de pertes de paquets allant jusqu'à 70%, le robot garde une bonne performance, il arrive à sa position finale tout en gardant une dynamique stable, le comportement est similaire à celui de la figure 5.3. Si le taux de pertes de paquets dépasse les 70% et jusqu'à 90%, la performance du robot se dégrade, la réponse du système présente des dépassements pour les positions X et Y du robot. Le robot met plus de temps à arriver à sa destination, pour le même temps de simulation le robot ne se stabilise pas dans sa position finale. Le robot se stabilise après $150s$ au lieu de $20s$ (pour 80% de pertes) comme le montre la figure 5.7. Pour un taux qui dépasse les 90%, la QoC est mauvaise, l'erreur de position ne diminue pas, c'est-à-dire que le robot n'évolue pas vers sa destination, on constate une instabilité tout au long de sa trajectoire. En observant les réponses du système, en position et en orientation, des oscillations continues persistent pour la même durée de simulation. En augmentant le temps de simulation à $2000s$, le système reste instable (voir figure 5.8). Quand il reçoit une nouvelle commande, il la garde jusqu'à la réception d'une nouvelle valeur. L'évolution du robot est alors circulaire puisqu'il ne reçoit pas de nouvelles consignes qui lui permettent d'évoluer vers sa destination $(1, 1)$. Un autre défaut réseau, un retard de transmission de $2s$, a été introduit. Ce retard ne satisfait pas la contrainte décrite dans l'inégalité 2.1 qui garantit une bonne QoS : le retard dépasse la période d'échantillonnage ($T_e = 400ms$).

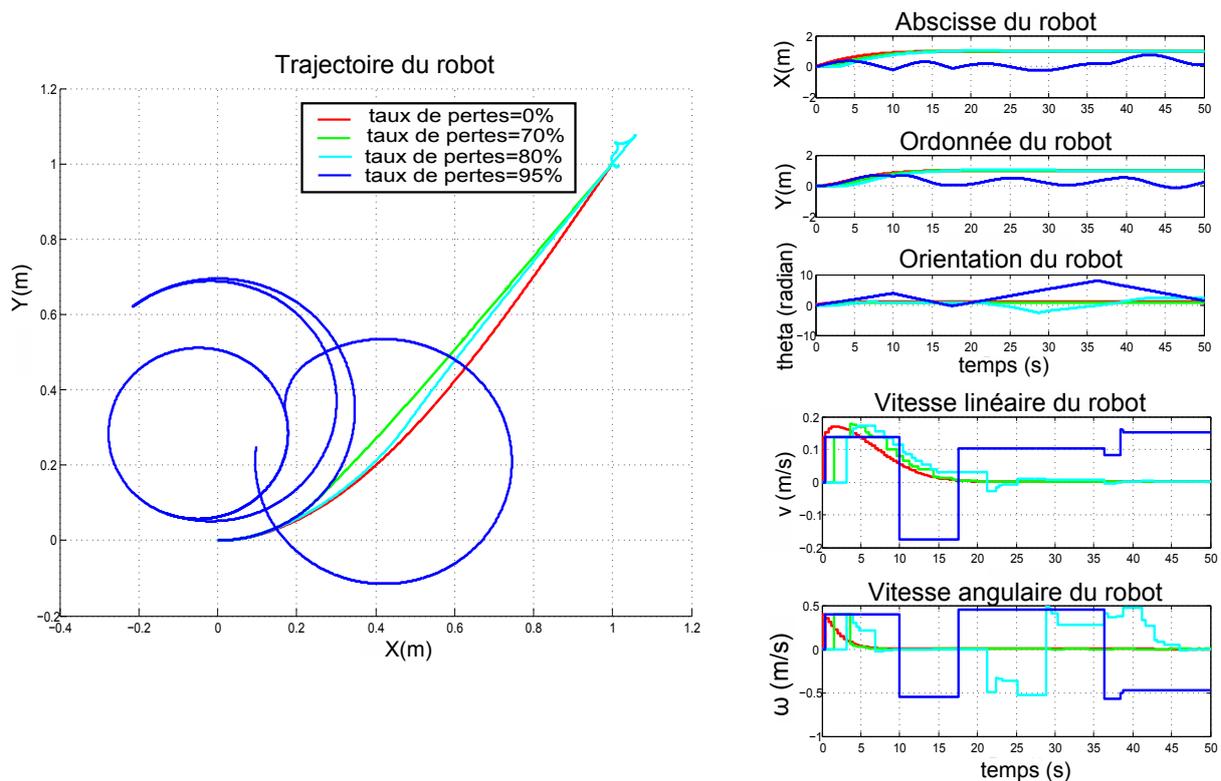


FIGURE 5.6 – Comportement du robot en fonction du taux de pertes de paquets

Donc, les paquets de commande "ratent" leurs instants de réception. La figure 5.9 montre que le robot arrive à sa destination de coordonnées (1,1) mais il ne stabilise pas son orientation avec de légers déplacements autour de sa position finale. Dans le cas où le réseau de communication est monopolisé par une autre application gourmande en matière de bande passante, le taux de pertes de paquets et le retard vont augmenter (cela va être étudié par la suite) et dégrader le comportement du robot allant vers sa destination.

5.3 Réseau bayésien pour le diagnostic : étude qualitative

Le système est composé d'un robot, équipé d'actionneurs et de capteurs, communiquant avec une station distante. Notre modélisation est le résultat de l'étude effectuée sur les défaillances du SCR qui ont un impact sur la QoC du système en terme de stabilité et de temps de réponse, c'est-à-dire la capacité du contrôleur à amener le robot à la position d'équilibre via le réseau tout en évitant les obstacles. La QoC est affectée par l'état du réseau, ce dernier étant un composant du système global, par l'état du robot et par

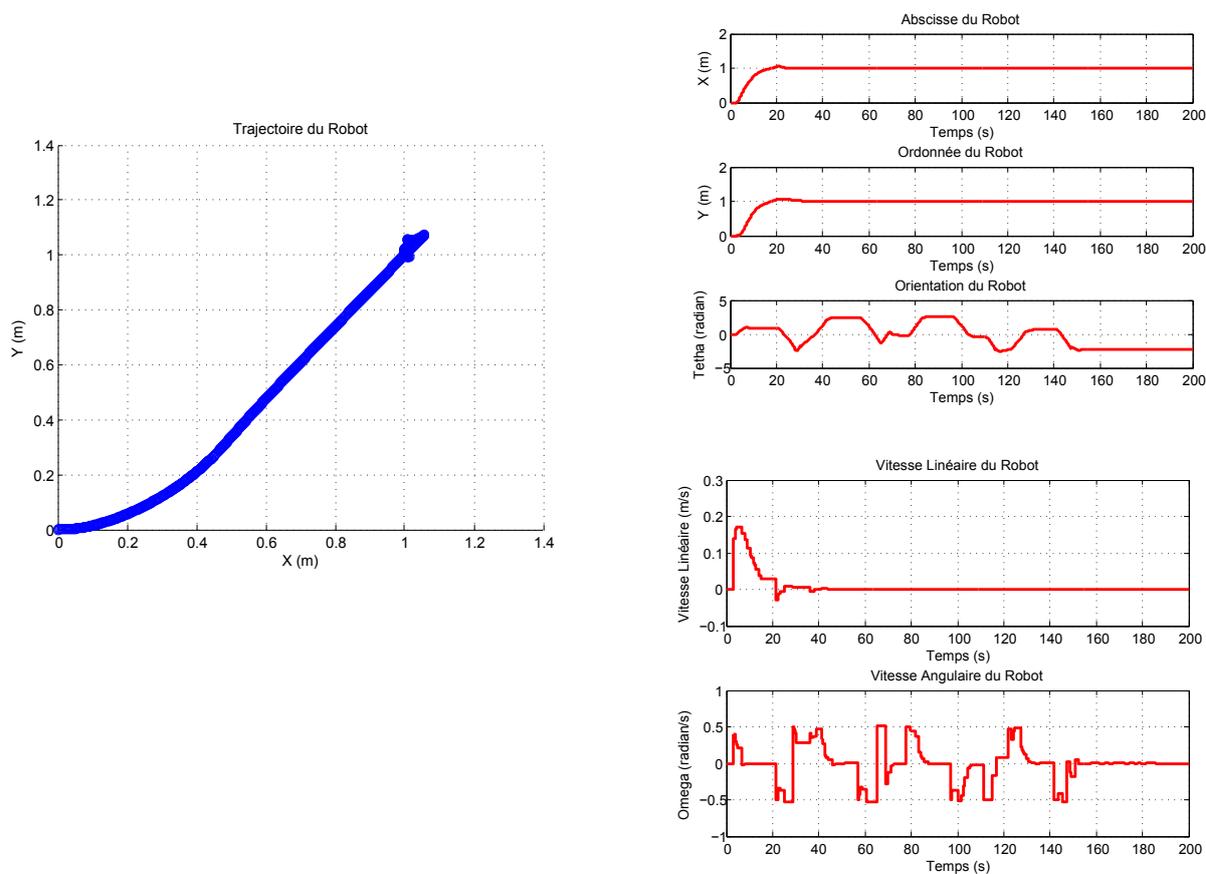


FIGURE 5.7 – Comportement du robot avec un taux de pertes de paquets de 80%

l'état du contrôleur. L'état du robot influence l'état de la QoC parce que dans le cas de la dégradation des actionneurs, ils ne sont plus capables d'exécuter les commandes, d'où la dégradation de la QoC. Si le contrôleur ne réussit pas à calculer les lois de commandes, la QoC va également se dégrader.

L'objectif de notre méthode de diagnostic est d'identifier la cause de dégradation de la QoC : soit une défaillance réseau, soit une défaillance au niveau du robot, soit une défaillance au niveau du contrôleur. On divise le système global en trois sous-systèmes : partie communication, partie opérative (actionneurs et capteurs) et partie commande. Le robot mobile a des limites en terme de capacité de calcul de son CPU, de sa mémoire et la courte durée de vie de sa batterie. En outre, la station et le robot n'ont pas la même vision du système global, chacun a des informations spécifiques sur l'état d'une des trois parties du système global. Pour toutes ces dernières raisons et pour ne pas surcharger le réseau avec les données nécessaires pour le diagnostic, nous avons décidé d'établir une méthode de diagnostic distribué où des agents collaborent pour un même objectif : déterminer la cause de la dégradation de la QoC. C'est une méthode de diagnostic probabiliste où un RB modulaire est implémenté. Ce RB modulaire est constitué de trois modules bayésiens cor-

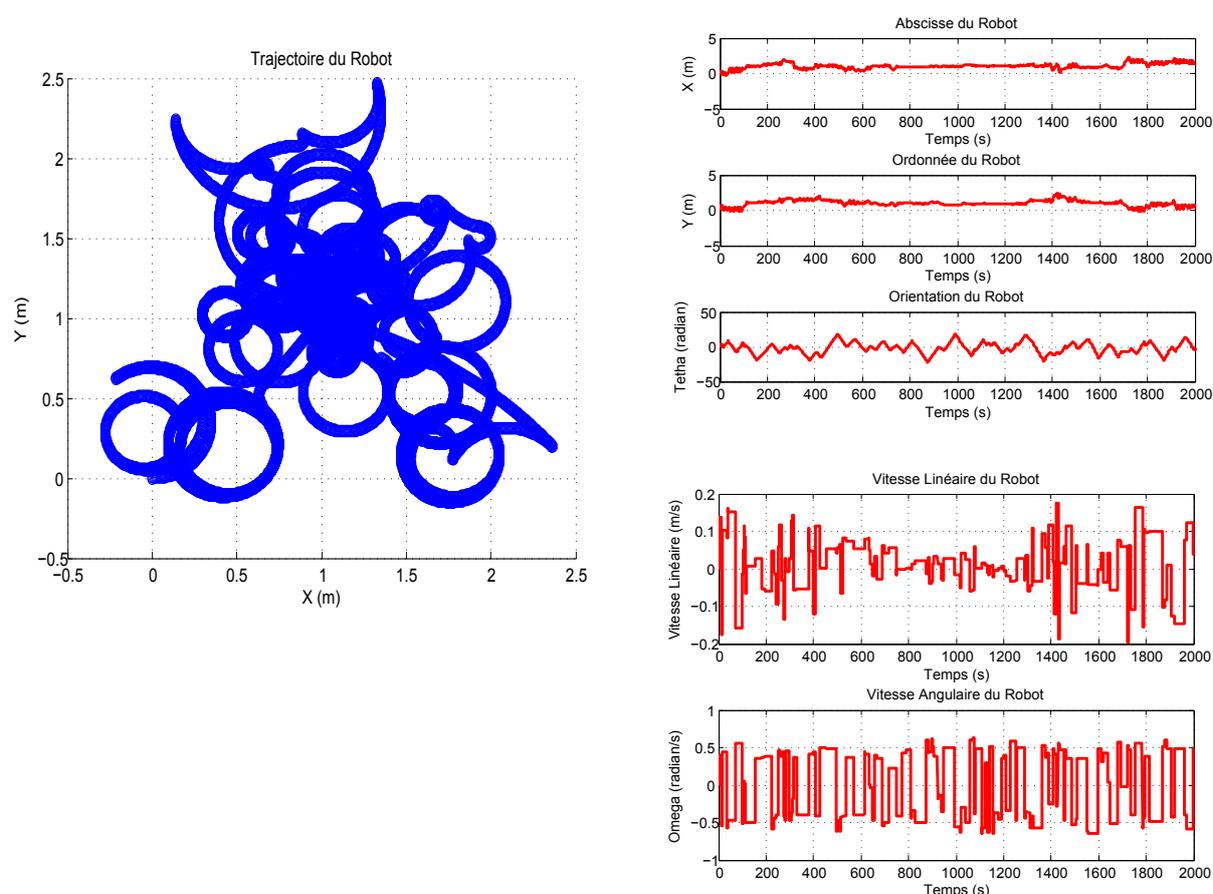


FIGURE 5.8 – Comportement du robot avec un taux de pertes de paquets de 95%

respondant chacun à une partie du système global comme le montre la figure 5.10. Le RB de la figure 5.10 représente les dépendances inter-domaines et intra-domaine d'où les liens causaux entre les modules bayésiens. Ces liens de cause-à-effet entre la partie communication et la partie commande d'une part et entre la partie opérative et la partie commande d'autre part sont représentés par les nœuds partagés *QoS* et *état du robot*. L'état de la partie commande est représenté par le nœud *QoC*, l'état de la partie communication est associé au nœud *QoS* et celui de la partie opérative est associé au nœud *état du robot*. Les modules bayésiens communiquent via les nœuds partagés par le biais de liens virtuels comme le montre la figure 5.10. Les modules bayésiens sont répartis entre les différents composants du système global. Les modules bayésiens de communication sont implémentés sur la station parce que la station a une vision globale sur tout le réseau de communication et les informations sur l'état de la mission du robot puisque le contrôleur est implémenté sur elle. Quant au module bayésien de la partie opérative, il est implémenté sur le robot parce que seul le robot peut surveiller son état interne : l'état de ses actionneurs, de ses capteurs et de sa batterie.

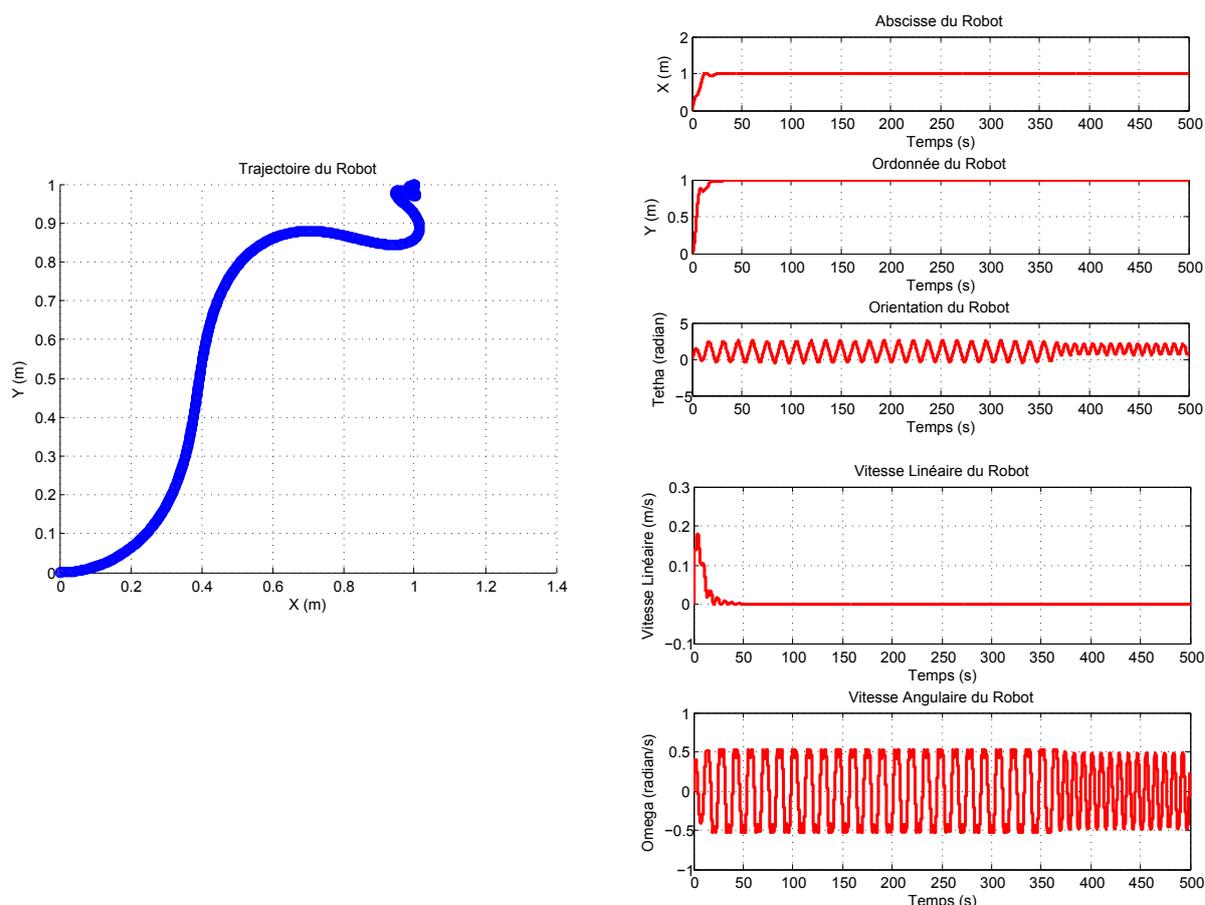


FIGURE 5.9 – Comportement du robot avec un retard de 2s

Dans un premier temps, la communication inter-modulaire consiste à transmettre les probabilités marginales a priori des nœuds partagés *QoS* et *état du robot* du module de communication et de la partie opérative au module de la partie commande. Dans un deuxième temps, la communication inter-modulaire procède dans le sens contraire pour le diagnostic. Il est alors possible de changer la variable d'entrée et de sortie pour l'inférence dans un module bayésien selon la situation. Tout cela montre l'intérêt et la flexibilité de l'utilisation du modèle bayésien modulaire de [LYC15], [LLC16] par rapport à celui de [Oud+10] qui ne permet la communication que dans un seul sens hiérarchique. Un autre avantage de l'utilisation de la structure modulaire bayésienne est la possibilité d'intégrer d'autres modules bayésiens de commande quand de nouveaux robots commencent à partager et à communiquer avec la station de contrôle. En outre, dans le cas d'une rupture de la connexion entre un robot et la station de contrôle, il est plus simple à mettre à jour l'architecture de diagnostic modulaire que l'architecture centralisée. Il suffit que la station de contrôle supprime le module bayésien de commande correspondant à ce robot et le robot commute sur un diagnostic embarqué où un RB monolithique est implémenté. Tout cela va être détaillé

dans la section 5.6.

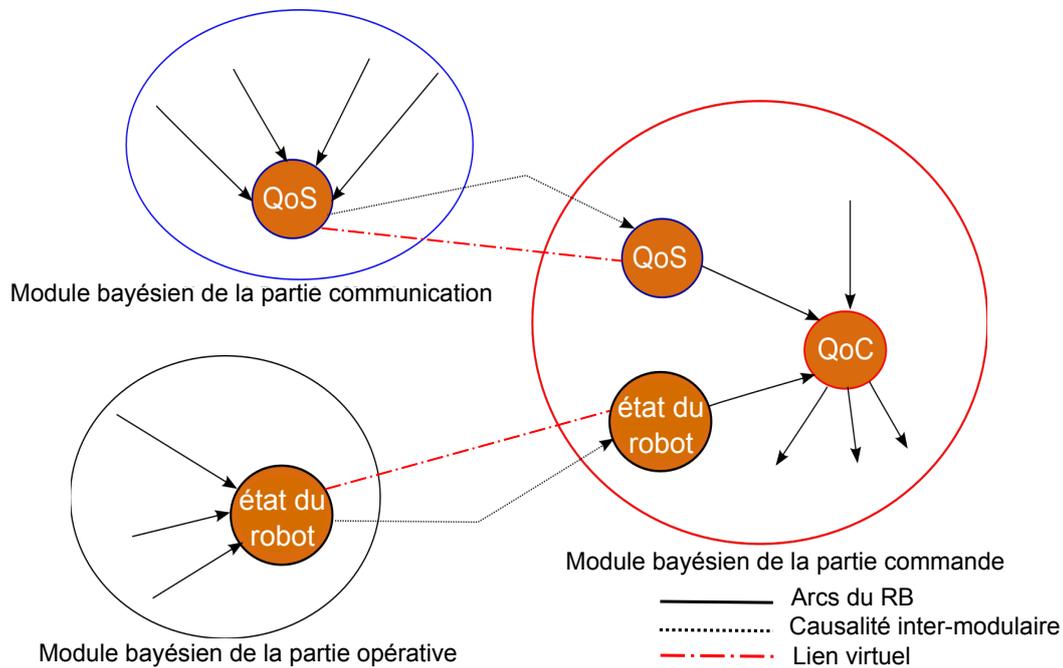


FIGURE 5.10 – Extrait du réseau bayésien modulaire pour le diagnostic d'un robot commandé en réseau

Chaque module bayésien est structuré suivant un modèle en couches comme celui de [PC07]. Ce modèle est utilisé pour avoir plus de précision et de couverture dans la modélisation des dépendances entre les nœuds, surtout ceux de la couche intermédiaire qui représentent les dépendances entre les défaillances des composants. La première couche dans ce modèle est la couche *usage* dont les nœuds représentent l'environnement et les conditions dans lesquels le système fonctionne. Une deuxième couche intermédiaire est la couche *composants* où les états de défaillances sont représentés. Une couche *sous-système* (facultative) représente l'état du sous-système qui est l'ensemble des composants. Finalement, une couche *observations* est utilisée pour surveiller l'état des composants et leurs défaillances. Dans la suite, nous allons décrire la construction et la définition des nœuds de chaque module bayésien structuré suivant ce modèle en couches.

5.3.1 Module bayésien de la partie commande

Le nœud *QoC*, qui représente l'état de la partie commande, indique la performance et l'état de l'application de contrôle-commande, c'est-à-dire si le système est commandable via le réseau ou non. La *QoC* est influencée par l'état du robot (état de la partie opérative),

par l'état du réseau (état de la partie de communication) et par l'état du contrôleur. En effet, si les moteurs (les actionneurs) n'arrivent pas à exécuter les commandes, le robot n'est pas capable d'atteindre la position finale. En outre, si le réseau ne réussit pas à fournir la QoS requise, le robot ne va pas recevoir les commandes au bon moment et ne sera pas capable d'envoyer de nouvelles mesures de ses capteurs. Enfin pour le contrôleur,

TABLE 5.1 – Définition des nœuds de la partie commande

Nœuds	Etats
QoC	bonne, dégradée, mauvaise
QoS	bonne, dégradée, mauvaise
état du robot	bon, dégradé, mauvais
contrôleur	bon, dégradé, mauvais
distance robot-obstacle	$distance \geq rayon_{sécurité} = 13.5cm, distance < rayon_{sécurité}$
stabilité	stable, non stable
erreur de position	croissante, décroissante
type de contrôleur	atteinte de cible, évitement d'obstacle
charge CPU	charge CPU $\geq s, charge CPU < s$

si le calculateur n'arrive pas à calculer les lois de commandes, le robot ne va plus recevoir les données nécessaires au bon déroulement de sa mission. Les trois nœuds *QoS*, *état du robot* et *contrôleur* ont alors un lien causal avec le nœud *QoC* comme le montre la figure 5.11. Pour observer l'état de la QoC, trois nœuds de type *observation* sont définis :

- nœud *stabilité* : ce nœud évalue la stabilité du système le long de la trajectoire en utilisant la théorie de Lyapunov ;
- nœud *erreur de position* : ce nœud surveille la mission de navigation du robot (le robot doit atteindre la cible dans un temps fini) en évaluant la distance qui sépare le robot de sa destination. Cette distance doit diminuer tout au long de la mission de navigation ;
- nœud *distance robot-obstacle* : ce nœud surveille la mission d'évitement d'obstacle parce que le robot doit atteindre la cible en évitant les obstacles. Le robot doit rester en dehors de la zone de l'obstacle, c'est-à-dire à une distance supérieure à un rayon de sécurité défini selon l'obstacle.

La table 5.1 décrit tous les états possibles de chaque nœud de la partie commande. Le nœud *contrôleur* est un nœud *composant* qui a une influence sur l'état de la QoC. Si le contrôleur a une faible capacité de calcul ou si son CPU est saturé, on aura une dégradation de la QoC. On définit alors deux nœuds *usage* correspondant au nœud *contrôleur* :

- nœud *type de contrôleur* : c'est le contrôleur d'évitement d'obstacle qui est activé si le robot est en mission d'évitement d'obstacle sinon c'est le contrôleur d'atteinte de cible qui s'exécute. Le contrôleur d'évitement d'obstacles est plus sensible aux perte de données ;

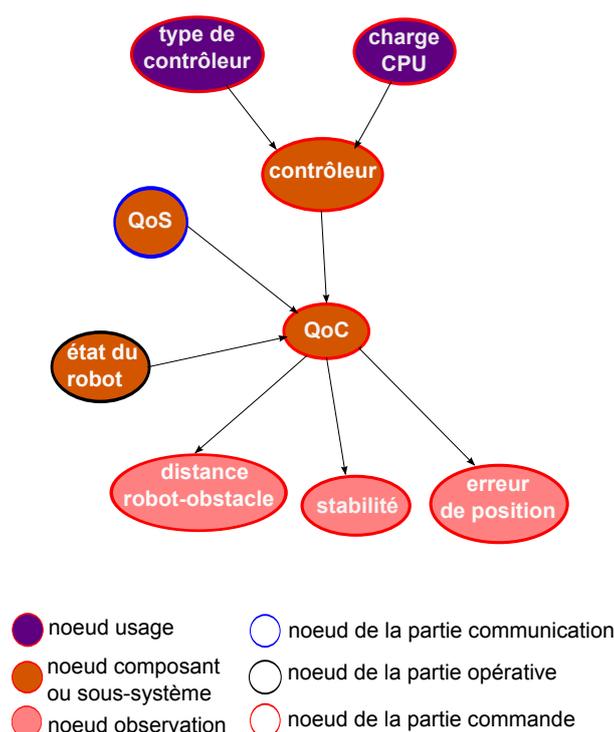


FIGURE 5.11 – Module bayésien de la partie commande

- noeud *charge CPU* : c'est la charge utilisée par les applications (donc les ressources allouées aux applications) qui s'exécutent sur le processeur du serveur des lois de commandes.

5.3.2 Module bayésien de la partie opérative

Le robot est équipé d'un ensemble de détecteurs d'obstacles et d'actionneurs qui constituent la partie opérative du SCR. C'est la partie déterministe du système. Un nœud de type *sous-système* qui est *état du robot* est défini pour indiquer l'état de cette partie comme le montre la figure 5.12. Les composants (nœuds *composants*) de cette partie sont : le moteur gauche, le moteur droit et les capteurs infra-rouges permettant de détecter les obstacles. Un nœud *observation* est associé à chaque nœud composant. Ce nœud a pour rôle de mesurer et de surveiller l'état de santé de chaque nœud composant. Un capteur est dans un état défaillant s'il transmet des valeurs erronées. Pour l'ensemble des capteurs détectant les obstacles (voir figure B.4 en annexe B), un coefficient de performance est défini pour vérifier la cohérence de toutes les mesures provenant des capteurs. Pour calculer ce coefficient, un poids est associé à chaque capteur et ce poids dépend de la position du capteur sur le robot et de son rôle dans l'évitement de l'obstacle. Ainsi, les capteurs {1, 8, 9} ont un poids

de 1, les capteurs $\{2, 3, 6, 7\}$ possèdent un poids de 2 et un poids de 3 est attribué aux capteurs $\{4, 5\}$. Un vecteur de poids P est alors défini décrivant le poids associé à chaque capteur allant de 1 à 9 tel que $P = [1 \ 2 \ 2 \ 3 \ 3 \ 2 \ 2 \ 1 \ 1]$. Pour tester la fiabilité des capteurs à détecter la présence d'un obstacle, un vecteur A des mesures transmises par les capteurs est défini et comparé à un vecteur B de valeurs référence de chaque capteur. Le coefficient de performance est alors déterminé à partir de l'équation 5.3 pour caractériser l'état bon, dégradé ou mauvais de l'ensemble du réseau des capteurs détectant les obstacles. Des seuils s_1 et s_2 de performances sont calculés à partir des modes de défaillance des capteurs qui ont un impact sur la capacité du robot à éviter l'obstacle [Leg17]. Le seuil s_2 représente la limite de l'état mauvais des capteurs qui est défini si un de ces cas se produit :

- le capteur 4 ou 5 est défectueux ;
- deux capteurs de l'ensemble $\{2, 3, 6, 7\}$ sont défectueux ;
- les capteurs 1, 8 et 9 sont tous les trois défectueux.

Le seuil s_1 , qui est la limite de la bonne performance, est déterminé de telle sorte qu'on a un état dégradé de l'ensemble des capteurs si l'un de ces cas survient :

- un des capteurs $\{2, 3, 6, 7\}$ est défectueux ;
- deux des capteurs $\{1, 8, 9\}$ sont défectueux.

On a un bon état de l'ensemble des capteurs si seulement si, au maximum, un des capteurs 1, 8 ou 9 est défectueux. Les seuils de performance des capteurs sont définis expérimentalement avec comme valeur 500 et 2000 pour correspondre à chaque état de l'ensemble des capteurs comme le montre la table 5.2. On utilise l'équation 5.3 pour calculer le coefficient de performance.

$$\text{coefficient de performance} = P * [A - B] \quad (5.3)$$

Pour les actionneurs, l'état du moteur est observable par l'erreur de consigne qui ne doit pas dépasser un seuil sinon le moteur est dans un état défaillant. L'état défaillant du moteur indique qu'il n'est pas capable d'exécuter les consignes et de répondre aux commandes appliquées. A chaque période d'échantillonnage $T_e = 400ms$, l'erreur de consigne e_k et la mesure de la vitesse angulaire ω_{mes} (voir figure 5.2) sont évaluées pour déterminer l'état de chaque moteur. En fonction du pourcentage d'erreur P_{erreur} entre la mesure ω_{mes} et la consigne $\omega_{ref,k}$, l'état du moteur est soit bon, soit dégradé, soit mauvais. P_{erreur} est calculé de la manière suivante :

$$P_{erreur} = 1 - \frac{\min(\omega_{mes}, \omega_{ref,k})}{\max(\omega_{mes}, \omega_{ref,k})} \quad (5.4)$$

Le robot arrive à destination et évite les obstacles sans difficultés quand le pourcentage P_{erreur} défini dans l'équation 5.4 est inférieur ou égal à 10%. Dans ce cas, les moteurs sont en bon état (voir figures 5.3 et 5.5). Quand l'état des moteurs est dégradé, pour $10\% < P_{erreur} \leq 30\%$, le robot atteint sa cible avec quelques oscillations au niveau de sa vitesse angulaire ω et une dégradation de la réponse de l'angle d'orientation θ au cours de sa trajectoire comme le montre la figure B.8 (en annexe B). Si $P_{erreur} > 30\%$, les moteurs sont en mauvais état, le robot arrive à destination avec un temps de réponse plus grand

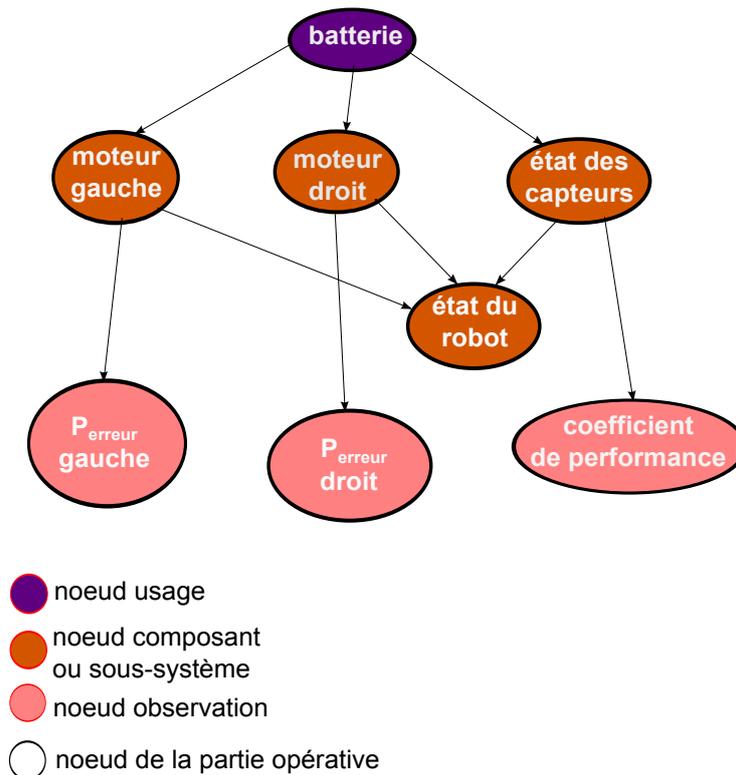


FIGURE 5.12 – Module bayésien de la partie opérative

(voir figure B.9 en annexe B). Pour le même temps d'expérimentation, le robot n'arrive pas à se stabiliser. Davantage de détails sur les expérimentations et les observations faites sur l'état des moteurs sont disponibles en annexe B dans la section B.3.

Un nœud *usage batterie* est défini pour représenter le mode d'utilisation de la batterie du robot. Le bon fonctionnement des capteurs et des moteurs est dépendant de la charge de la batterie. En effet, si la charge de la batterie n'est plus qu'à 10% de sa capacité, cela peut avoir une influence sur l'état des capteurs ou sur l'état des moteurs et ainsi impacter directement la QoC lors de la mission.

Si l'un des composants de la partie opérative tombe en panne (une défaillance physique), le système ne fonctionne plus et le composant doit être remplacé. La table 5.2 décrit tous les états possibles de chaque nœud de la partie opérative.

5.3.3 Module bayésien de la partie communication

Le réseau est un composant susceptible d'être défaillant comme les autres composants du système. Lorsque le réseau cesse d'assurer la QoS requise, il est considéré comme défaillant. Donc, l'état de la communication est représenté par l'état du nœud *QoS* qui est

TABLE 5.2 – Définition des nœuds de la partie opérative

Nœuds	Etats
état du robot	bon, dégradé, mauvais
moteur gauche	bon, dégradé, mauvais
moteur droit	bon, dégradé, mauvais
état des capteurs	bon, dégradé, mauvais
$P_{erreur,gauche}$	$P_{erreur} \leq 10\%$, $10\% < P_{erreur} \leq 30\%$ $P_{erreur} > 30\%$
$P_{erreur,droit}$	$P_{erreur} \leq 10\%$, $10\% < P_{erreur} \leq 30\%$ $P_{erreur} > 30\%$
coefficient de performance	coefficient ≤ 500 , $500 < \text{coefficient} \leq 2000$ coefficient > 2000
batterie	$batterie \geq 10\%$, $batterie < 10\%$

un nœud de type *sous-système* comme le montre la figure 5.13. La QoS désigne la capacité à fournir un support de communication conforme aux exigences en matière de temps et de bande passante et à garantir un niveau acceptable de pertes de paquets. D'une manière générale, les paramètres influençant la QoS sont les pertes de paquets, le délai, la gigue, le débit et le RSSI. Des défaillances, qui ont une influence sur l'état du réseau, c'est-à-dire l'état de sa QoS, sont associées à ces paramètres auxquels on associe des nœuds de type *observation*. Il existe deux types de défaillances dans un réseau de communication sans fil où les entités communicantes sont mobiles :

- les erreurs liées au protocole de communication : par exemple l'erreur de transmission peut être due à une collision de paquets elle-même due à un réseau surchargé ou à des périodes de transmission trop petites ;
- les erreurs liées à la couche physique : par exemple l'erreur du canal de communication peut être due à l'interférence ou à l'évanouissement du signal de transmission.

Pour les données qui circulent dans le réseau, les types de défaillances suivants sont définis pour représenter les différents problèmes de transmission :

- nœud *perte de données* : l'envoi des données via un réseau sans fil est susceptible d'avoir des pertes à cause d'erreurs au niveau de la couche physique, de défaillances au niveau de la liaison ou de congestion du réseau. Pour les systèmes temps réel, un long retard est considéré comme une perte de l'information. L'état de ce nœud est surveillé par le nœud *taux de pertes de paquets* qui donne le taux de pertes de paquets à chaque envoi de messages.
- nœud *respect du délai* : le délai dans un réseau sans fil est aléatoire et variable. Le SCR sans fil est un système temps réel qui fonctionne périodiquement et le non respect de la date limite d'arrivée à cause de la gigue ou bien d'un long délai de transmission entraîne la perte du paquet, d'où le lien causal entre les nœuds *respect*

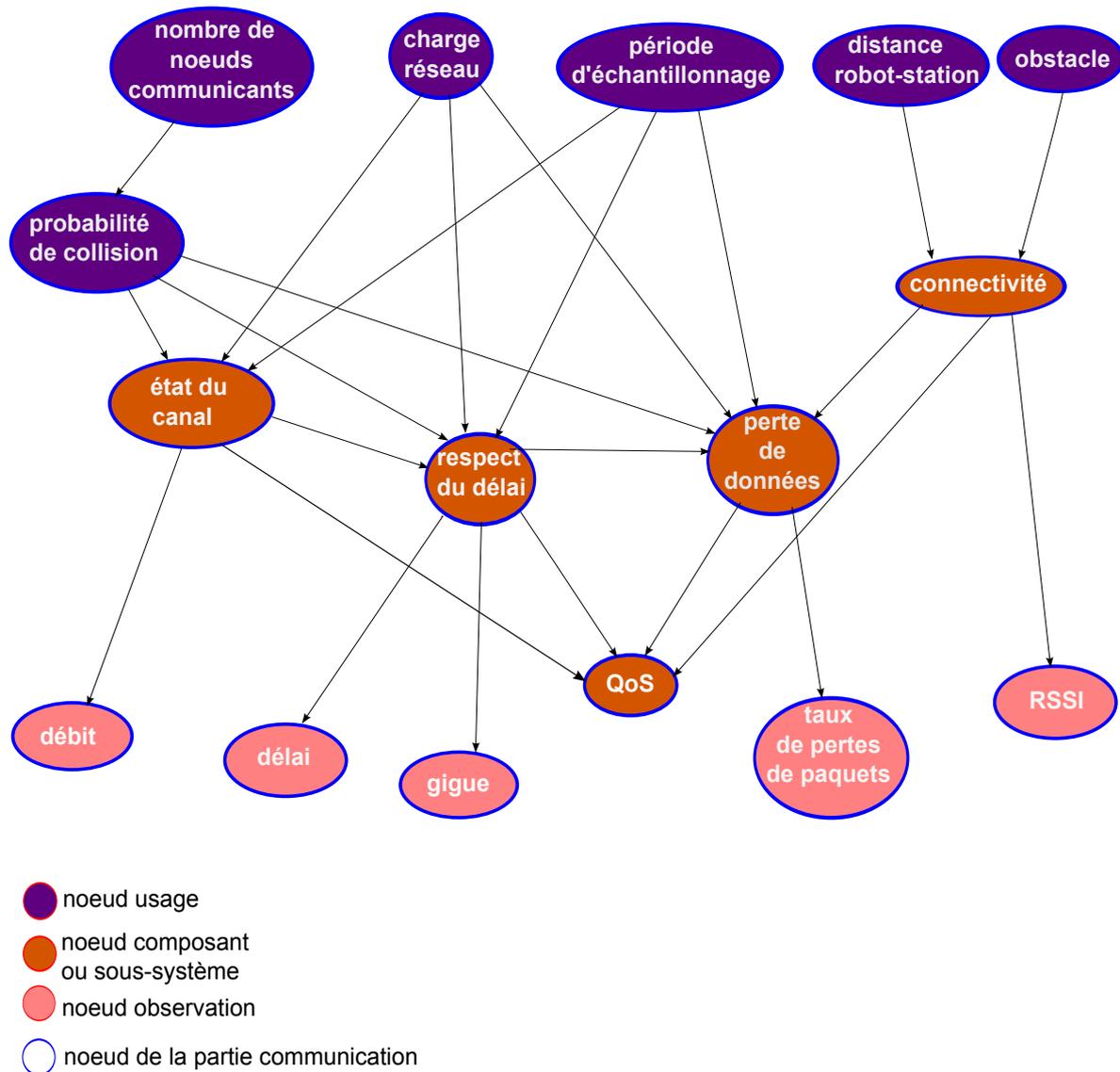


FIGURE 5.13 – Module bayésien de la partie communication

du délai et perte de données.

Le nœud *état du canal de communication* est un autre problème qui peut perturber la disponibilité du réseau en limitant la bande passante. Si le canal de communication est saturé, l'envoi de messages n'est pas garanti, ce qui va entraîner une dégradation de la QoS. Le nœud *débit*, défini comme un nœud *observation* pour le nœud *état du canal de communication*, mesure la quantité d'informations envoyée par seconde et il quantifie la limite de la bande passante dans le réseau. Un canal de communication saturé provoque de longs délais ce qui explique le lien causal entre les nœuds *état du canal de communication* et *respect du délai*. L'application a un débit minimal qui lui garantit une bonne performance

selon la période d'échantillonnage définie. Cette application ne doit pas consommer toute la bande passante sinon les autres applications vont souffrir d'une pénurie de bande passante. Le débit de l'application ne doit pas alors dépasser le débit de saturation défini dans l'équation 4.33 qui varie en fonction du nombre de boucles de commande qui partagent le réseau (voir figure 4.4).

Un autre type de défaillance qui a aussi un impact sur la performance du réseau est la dégradation de la connectivité. La perte de connectivité cause des erreurs de transmissions dans le canal de communication. Si le robot s'éloigne de la station et sort de sa zone de couverture, il va perdre la connexion avec la station, ce qui va engendrer une perte de messages, d'où le lien causal entre les nœuds *connectivité* et *perte de données*. Le nœud *RSSI*, le *Request Signal Strength Indication*, mesure la puissance du signal reçu et est associé, comme un nœud *observation*, au nœud *composant connectivité*.

Les conditions d'utilisation du réseau et l'environnement dans lequel la communication entre la station et le robot se déroule sont définis par l'ensemble des nœuds de la couche *usage* :

- nœuds *nombre de nœuds communicants* et *probabilité de collision* : ils définissent le nombre d'entités communicantes (donc le nombre de boucles de commandes) dans le réseau et la probabilité de collision de paquets respectivement. En augmentant le nombre de nœuds communicants, la probabilité d'avoir une collision augmente, d'où le lien causal entre ces deux nœuds du RB. Si la probabilité de collision augmente, le débit diminue, le retard augmente et le taux de pertes de paquets augmente aussi. C'est ainsi que les liens de cause-à-effet entre le nœud *probabilité de collision* et les nœuds *état du canal de communication*, *respect du délai* et *perte de données* sont établis ;
- nœud *charge réseau* : il représente la charge utilisée par les différentes applications qui partagent le réseau. Si le réseau devient surchargé, il perd ses performances, le retard augmente, le débit diminue et le taux de pertes de paquets augmente, d'où les liens causaux entre le nœud *charge réseau* et les nœuds *état du canal de communication*, *respect du délai* et *perte de données* ;
- nœud *période d'échantillonnage* : il représente la valeur de la période d'échantillonnage pour la boucle principale de commande qui ne doit pas être inférieure à une limite, sinon le réseau va être surchargé et les messages envoyés vont être perdus, d'où la dépendance causale entre le nœud *période d'échantillonnage* et les nœuds *état du canal de communication*, *respect du délai* et *perte de données* ;
- nœud *obstacle* : il définit le milieu dans lequel le robot navigue, soit un milieu avec présence d'obstacles, de murs de natures différentes (bois, plastique, béton, métal...), soit un milieu dégagé sans obstacle. La présence des obstacles et des murs entre la station et le robot provoque de l'évanouissement et de la dégradation du signal, d'où le lien de cause-à-effet entre les nœuds *obstacle* et *connectivité* ;
- nœud *distance robot-station* : il définit la zone de couverture de la station. Si le robot

dépasse cette zone, il perdra la connexion avec la station. La mobilité du robot et son éloignement de la station provoquent l'atténuation du signal et la dégradation de la connectivité (voir figure B.6 de l'annexe B), d'où la dépendance causale entre les nœuds *distance robot-station* et *connectivité*.

La table 5.3 décrit tous les états possibles de chaque nœud de la partie communication, où S correspond au débit de saturation qui varie en fonction du nombre de boucles de commande qui partagent le même réseau.

TABLE 5.3 – Définition des nœuds de la partie communication

Nœuds	Etats
connectivité	bonne, dégradée, mauvaise, perdue
perte de données	faible, moyenne, forte
respect du délai	oui, non
état du canal	saturé, non saturé
débit	débit $< 3.2kbits/s$, $3.2kbit/s \leq \text{débit} \leq S$, débit $> S$
délai	délai $\leq T_e$, délai $> T_e$
gigue	$gigue \leq gigue_{max}$, $gigue > gigue_{max}$
taux de perte de paquets	$0\% \leq \text{taux de perte de paquets} \leq 70\%$, $70\% < \text{taux de perte de paquets} < 90\%$, $90\% \leq \text{taux de perte de paquets} \leq 100\%$
RSSI	$RSSI > -80dBm$, $-85dBm \leq RSSI \leq -80dBm$, $-92dBm \leq RSSI < -85dBm$, $RSSI < -92dBm$
obstacle	oui, non
distance robot-station	distance robot-station $< 30m$, $30m \leq \text{distance robot-station} \leq 50m$, $50m < \text{distance robot-station} \leq 100m$ distance robot-station $> 100m$
période d'échantillonnage	période d'échantillonnage $\geq T_c$, période d'échantillonnage $< T_c$
charge réseau	charge réseau $< 4Mbit/s$, $4Mbit/s \leq \text{charge réseau} < 6Mbit/s$, charge réseau $\geq 6Mbit/s$
nombre de nœuds communicants	nombre de nœuds communicants < 95 , nombre de nœuds communicants ≥ 95
probabilité de collision	probabilité de collision ≤ 0.3 , probabilité de collision > 0.3

5.4 Réseau bayésien pour le diagnostic : étude quantitative

L'étude quantitative consiste à définir les lois de probabilités associées à chaque nœud du RB. Pour l'apprentissage des paramètres du RB, des scénarios de simulations sur Matlab et *Truetime* et des expérimentations sur un banc de tests ont été effectués. Le banc de tests et les expérimentations ont été l'objet des travaux de stage [Leg17] qui sont décrits dans l'annexe B. Le banc de tests est composé d'un robot Khepera 3 et d'une machine qui le contrôle. Quant au modèle du robot simulé, un bloc *kernel* de *Truetime* est associé à chaque partie de la boucle de commande du SCR sans fil : contrôleur, actionneurs et capteurs. En outre, un bloc *network* simule le comportement du réseau sans fil IEEE 802.11b où les paramètres du réseaux peuvent être configurés. Les probabilités marginales et conditionnelles de toutes les variables des modules bayésiens sont obtenues en utilisant l'équation 3.1.

5.4.1 Partie commande

Pour définir les probabilités a priori du RB de la partie commande de la figure 5.11, une étude statistique est faite afin de quantifier les relations de dépendances entre les nœuds. Pour certaines probabilités, l'avis d'un expert et l'étude statistique n'ont pas permis de les déterminer. Un apprentissage en ligne est alors utilisé. Pour les nœuds de la couche *usage : charge CPU*, nous supposons que si le serveur de commandes est saturé où $charge\ CPU > s$, le contrôleur devient mauvais comme le montre la table 5.4, avec b=bonne, dé=dégradée et m=mauvaise. Pour définir les trois états de la QoC, l'erreur de position d , la stabilité

TABLE 5.4 – Table de probabilités conditionnelles de l'état du contrôleur

charge CPU	$\leq s$		$> s$	
	cible	évitement	cible	évitement
contrôleur=b	0.874	0.632	0	0
contrôleur=dé	0.122	0.172	0	0
contrôleur=m	0.004	0.196	1	1

le long de la trajectoire et la distance qui sépare le robot d'un obstacle sont surveillés. Ces paramètres représentent les nœuds *observation* du nœud *QoC*, ils vont alors permettre de détecter la dégradation de la QoC. La stabilité est évaluée en étudiant l'évolution de la fonction de Lyapunov V , définie dans l'équation B.19 de l'annexe B. Pour garantir la stabilité du système le long de sa trajectoire, l'inégalité 5.5 doit être vérifiée, ce qui implique

que la dérivée de la fonction de Lyapunov soit définie négative :

$$\frac{V_{t_{k+1}} - V_{t_k}}{t_{k+1} - t_k} < 0 \quad (5.5)$$

où t_{k+1} et t_k sont les instants d'échantillonnage et $V_{t_{k+1}}$ et V_{t_k} sont les valeurs de la fonction de Lyapunov aux instants d'échantillonnage. Pour chaque état des nœuds de la couche *observation* de la partie commande (figure 5.11), un état du nœud *QoC* est défini. Les états du nœud *QoC* sont déterminés comme suit, où d représente l'erreur de position et \dot{d} est la variation de l'erreur de position.

Algorithm 1 Définition des états de la *QoC*

```

1: if  $d == 0$  &&  $\dot{d} == 0$  then
2:    $QoC \leftarrow$  bonne
3: else
4:   if  $\dot{d} < 0$  then
5:     if stabilité==1 then
6:        $QoC \leftarrow$  bonne
7:     else
8:        $QoC \leftarrow$  dégradée
9:     end if
10:  else
11:     $QoC \leftarrow$  mauvaise
12:  end if
13: end if

```

En surveillant l'état des missions d'atteinte de cible et d'évitement d'obstacles en fonction de l'état de la *QoS* (bonne, dégradée, mauvaise) du réseau et de l'état du robot (bon, dégradé, mauvais), on définit les probabilités conditionnelles du nœud *QoC* sachant les états de ses parents dans les tables 5.5, 5.6 et 5.7. Nous supposons que, dans le cas où le CPU est surchargé, le serveur de commandes n'est plus capable de fournir les nouvelles valeurs de commandes. L'état du nœud *QoC* devient alors mauvais quel que soit l'état des nœuds *QoS* et *état du robot*. Il est, par ailleurs, dans un état dégradé quand l'état du nœud contrôleur est dégradé. Les probabilités marginales a priori des nœuds racines dans les modules bayésiens et partagés avec d'autres modules suivent une loi uniforme [LYC15]. Pour plus de pertinence, les probabilités marginales des nœuds *QoS* et *état du robot* sont définies à partir des résultats d'inférence des modules bayésiens de la partie communication et de la partie opérative respectivement. Cela va être détaillé dans la section 5.5 avec des scénarios de fonctionnement.

TABLE 5.5 – Table de probabilités conditionnelles de l'état de la QoC sachant un bon état du contrôleur

contrôleur	bon								
état du robot	b			dé			m		
QoS	b	dé	m	b	dé	m	b	dé	m
QoC=b	0.876	0.767	0	0.859	0.744	0	0.568	0.537	0
QoC=dé	0.123	0.215	0.17	0.105	0.194	0	0.308	0.338	0.22
QoC=m	0.001	0.018	0.83	0.036	0.062	1	0.124	0.125	0.78

TABLE 5.6 – Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur dégradé

contrôleur	dégradé								
état du robot	b			dé			m		
QoS	b	dé	m	b	dé	m	b	dé	m
QoC=b	0	0	0	0	0	0	0	0	0
QoC=dé	1	1	0	1	1	0	0	0	0
QoC=m	0	0	1	0	0	1	1	1	1

TABLE 5.7 – Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur mauvais

contrôleur	mauvais								
état du robot	b			dé			m		
QoS	b	dé	m	b	dé	m	b	dé	m
QoC=b	0	0	0	0	0	0	0	0	0
QoC=dé	0	0	0	0	0	0	0	0	0
QoC=m	1	1	1	1	1	1	1	1	1

5.4.2 Partie opérative

Pour définir les probabilités conditionnelles correspondant au RB de la partie opérative de la figure 5.12, des scénarios de test ont été utilisés. Ils ont permis de définir l'état du robot à partir des états de ses composants : les moteurs et les capteurs de détection d'obstacles selon le niveau d'utilisation de la batterie. La table 5.8 est le résultat de cette étude où les probabilités conditionnelles a priori du nœud état du robot sont définies sachant les états de ses nœuds parents : *moteur droit*, *moteur gauche* et *état des capteurs*. Pour le cas où l'état du nœud *état des capteurs* est dégradé ou mauvais, nous considérons que le robot est en état dégradé ou mauvais respectivement.

Les probabilités conditionnelles a priori des états du moteur gauche/droit sachant l'état

TABLE 5.8 – Table de probabilités conditionnelles de l'état du robot

état des capteurs	bon								
	b			dé			m		
MD	b	dé	m	b	dé	m	b	dé	m
MG	b	dé	m	b	dé	m	b	dé	m
R=b	1	0	0	0	0	0	0	0	0
R=dé	0	1	0	1	1	0	0	0	0
R=m	0	0	1	0	0	1	1	1	1

de la batterie sont déterminées à partir du taux de défaillance donné par le constructeur. La table de probabilités conditionnelles de l'état du moteur sachant l'état de la batterie, définie dans la table 5.9, est établie à partir des données définies dans [RMT12]. L'avis

TABLE 5.9 – Table de probabilités conditionnelles de l'état du moteur

batterie	$\geq 10\%$	$< 10\%$
MG/MD=b	0.98	0
MG/MD=dé	0.0195	0
MG/MD=m	0.0005	1

de l'expert ou du constructeur pour les probabilités conditionnelles a priori de l'état des capteurs sachant l'état de la batterie ne sont pas disponibles. Les données correspondant au taux d'utilisation de la batterie sont également manquantes. Par conséquent, un apprentissage en ligne est effectué pour définir les probabilités de la table 5.11. Les probabilités conditionnelles définies dans la table 5.10 supposent que l'état des capteurs est bon si la charge de la batterie est supérieure à 10%.

TABLE 5.10 – Table de probabilités conditionnelles de l'état des capteurs

batterie	$\geq 10\%$	$< 10\%$
état de capteurs=b	1	0
état de capteurs=dé	0	0
état de capteurs=m	0	1

TABLE 5.11 – Table de probabilités a priori de l'état de la batterie

état du nœud	probabilités
<i>batterie</i> $\geq 10\%$	0.86
<i>batterie</i> $< 10\%$	0.14

5.4.3 Partie communication

Le réseau de communication sans fil est la source de l'indéterminisme. Dans l'étude faite dans le chapitre 4, nous avons défini les paramètres de la QoS qui permettent d'obtenir un comportement déterministe du réseau, comme le montrent les tables de probabilités 5.12, 5.13 et 5.14 en choisissant un usage en fonction :

- du nombre de robots qui partagent le même réseau et la même station de contrôle ;
- de la probabilité de collision qui garantit un délai qui ne dépasse les $400ms$;
- de la période d'échantillonnage de la boucle de commande dans le réseau choisie selon les ressources réseau disponibles.

Cette partie du module bayésien de la partie communication modélise un réseau dont le comportement est déterministe. Les seuils de fonctionnement déterministes sont définis grâce à l'analyse faite dans le chapitre 4. Pour avoir une période d'échantillonnage minimale de $0.4s$ (cette valeur correspond à la condition définie dans l'inégalité 5.1), le nombre de boucles de commande partageant le même réseau ne doit pas dépasser 95 boucles (voir figure 4.5). En d'autres termes, la station de contrôle peut accepter jusqu'à 95 robots. Ce nombre de boucles de commande correspond à une probabilité de collision maximale de 0.3 pour une taille de fenêtre de contention minimale CW_{min} égale à 32 (voir figure 4.3). Par conséquent, au-delà de ces seuils en terme de nombre de nœuds communicants et de probabilité de collision, le niveau de la QoS ne satisfait pas les exigences de l'application. Le délai, le débit et le taux de pertes de paquets augmentent engendrant une mauvaise QoC.

TABLE 5.12 – Table de probabilités conditionnelles du nœud respect du délai sachant l'état du nœud probabilité de collision

probabilité de collision	≤ 0.3	> 0.3
respect du délai = <i>oui</i>	1	0
respect du délai = <i>non</i>	0	1

TABLE 5.13 – Table de probabilités conditionnelles du nœud état du canal sachant l'état du nœud probabilité de collision

probabilité de collision	≤ 0.3	> 0.3
état du canal = non saturé	1	0
état du canal = saturé	0	1

Cependant, le réseau est susceptible d'être surchargé par d'autres applications gourmandes

TABLE 5.14 – Table de probabilités conditionnelles du nœud pertes de données sachant l'état du nœud probabilité de collision

probabilité de collision	≤ 0.3	> 0.3
perte de données = <i>faible</i>	1	0
perte de données = <i>moyenne</i>	0	0
perte de données = <i>forte</i>	0	1

en terme de bande passante, voire par des applications vulnérables et des cyberattaques. La charge du réseau doit alors être surveillée. Sa variation engendre un comportement stochastique : la transmission et la réception des paquets ne sont plus garanties. Ce comportement est représenté dans les tables de probabilités conditionnelles des nœuds *perte de données*, *respect du délai* et *état du canal* (tables 5.15, 5.16, 5.17). Dans un scénario d'expérimentation, la charge maximale du réseau a été définie selon les trois intervalles du taux de pertes de paquets définis dans la table 5.3. Une application, qui partage le même réseau que le robot et la station de contrôle, est responsable de la surcharge du réseau 802.11b avec un débit de 11Mbit/s. Le débit de l'application de commande est estimé à 3.2kbit/s. Cette valeur a été trouvée en utilisant l'équation 4.34 du chapitre 4 en prenant une période d'échantillonnage égale à 0.4s. Cette valeur est validée par le banc de tests décrit en annexe B.

TABLE 5.15 – Table de probabilités conditionnelles du nœud pertes de données sachant l'état de charge du réseau

charge réseau	$charge < 4Mbit/s$	$4Mbit/s \leq charge < 6Mbit/s$	$charge \geq 6Mbit/s$
perte de données = <i>faible</i>	1	0	0
perte de données = <i>moyenne</i>	0	1	0
perte de données = <i>forte</i>	0	0	1

Les probabilités des nœuds observations correspondant aux différents nœuds de type *composant* sont présentées dans les tables 5.18, 5.19, 5.20, 5.21. La valeur de la gigue acceptable, qui est égale à 398ms est déterminée à partir de l'équation 4.41 du chapitre 4 avec $T_e = 400ms$ et $T_{avgRTT} = 2ms$. Ces tables montrent que pour chaque état du nœud

TABLE 5.16 – Table de probabilités conditionnelles du nœud respect du délai sachant l'état de charge du réseau

charge réseau	$charge < 4Mbit/s$	$4Mbit/s \leq charge < 6Mbit/s$	$charge \geq 6Mbit/s$
respect du délai = <i>oui</i>	0.7	0.12	0.04
respect du délai = <i>non</i>	0.3	0.88	0.96

TABLE 5.17 – Table de probabilités conditionnelles du nœud état du canal sachant l'état de charge du réseau

charge réseau	$charge < 4Mbit/s$	$4Mbit/s \leq charge < 6Mbit/s$	$charge \geq 6Mbit/s$
état du canal = non saturé	1	0	0
état du canal = saturé	0	1	1

de type *composant*, un état du nœud de type *observation* lui est associé. Chaque nœud de type *observation* représente un paramètre de la QoS. Le paramètre est évalué et l'état du nœud de type *observation* est identifié par rapport à la valeur rendue par l'évaluation du paramètre.

TABLE 5.18 – Table de probabilités conditionnelles du nœud taux de pertes de paquets

perte de données	faible	moyenne	forte
$0\% \leq \text{taux de pertes} \leq 70\%$	1	0	0
$70\% < \text{taux de pertes} < 90\%$	0	1	0
$90\% \leq \text{taux de pertes} \leq 100\%$	0	0	1

TABLE 5.19 – Table de probabilités conditionnelles du nœud délai

respect du délai	oui	non
délai $\leq 400ms$	1	0
délai $> 400ms$	0	1

En outre, la mobilité des robots autour de la station de contrôle engendre d'autres perturbations liées à l'éloignement géographique et à l'atténuation du signal. Par conséquent, d'autres sources d'indéterminisme sont introduites influençant la connectivité comme dans le cas des probabilités conditionnelles du nœud *connectivité* sachant l'état des nœuds *obstacle* et *distance robot-station* représentées dans la table 5.22. L'impact de la dégradation

TABLE 5.20 – Table de probabilités conditionnelles du nœud gigue

respect du délai	oui	non
gigue $\leq 398ms$	1	0
gigue $> 398ms$	0	1

TABLE 5.21 – Table de probabilités conditionnelles du nœud débit

état du canal	non saturé	saturé
débit $< 3.2kbit/s$	0	0.999
$3.2kbit/s \leq \text{débit} \leq S$	1	0
débit $> S$	0	0.001

de la connectivité sur la transmission de données est défini à partir de l'état du nœud *perte de données* sachant l'état du nœud *connectivité* dans la table 5.24. L'état de la connectivité est observé grâce au *RSSI*, et les états du nœud *RSSI* sont définis dans la table 5.23.

TABLE 5.22 – Table de probabilités conditionnelles du nœud connectivité

distance robot-station	$\leq 30m$		$> 30m$ et $< 50m$		$\geq 50m$ et $\leq 100m$		$> 100m$	
	oui	non	oui	non	oui	non	oui	non
connectivité=b	0.44	0.96	0.11	0.46	0.06	0.14	0	0
connectivité=dé	0.40	0.04	0.53	0.29	0.30	0.38	0	0
connectivité=m	0.16	0	0.36	0.25	0.64	0.48	0	0
connectivité=perdue	0	0	0	0	0	0	1	1

TABLE 5.23 – Table de probabilités conditionnelles du nœud RSSI sachant l'état du nœud connectivité

connectivité	bonne	dégradée	mauvaise	perdue
$RSSI > -80dBm$	1	0	0	0
$-85dBm \leq RSSI \leq -80dBm$	0	1	0	0
$-92dBm \leq RSSI < -85dBm$	0	0	1	0
$RSSI < -92dBm$	0	0	0	1

Finalement, pour quantifier les états du nœud *QoS* (voir tables 5.25 à 5.28) les états des nœuds *connectivité*, *respect du délai* et *perte de données* sont pris en compte. Nous supposons que si l'état du nœud *état du canal* est saturé, l'état du nœud *QoS* est mauvais. En outre, nous supposons que le nœud *QoS* a le même état que son nœud parent *perte de données* quand les autres nœuds parents sont en bon état.

TABLE 5.24 – Table de probabilités conditionnelles du nœud pertes de données

connectivité	bonne	dégradée	mauvaise	perdue
perte de données = <i>faible</i>	1	0.58	0.16	0
perte de données = <i>moyenne</i>	0	0.42	0.83	0
perte de données = <i>forte</i>	0	0	0.01	1

TABLE 5.25 – Table de probabilités conditionnelles du nœud QoS pour une connectivité bonne

connectivité	bonne					
	oui			non		
respect du délai	faible	moyenne	forte	faible	moyenne	forte
perte de données = <i>faible</i>	1	0	0	0	0	0
QoS=bonne	1	0	0	0	0	0
QoS=dégradée	0	1	0	0	0	0
QoS=mauvaise	0	0	1	1	1	1

TABLE 5.26 – Table de probabilités conditionnelles du nœud QoS pour une connectivité dégradée

connectivité	dégradée					
	oui			non		
respect du délai	faible	moyenne	forte	faible	moyenne	forte
perte de données = <i>faible</i>	0	0	0	0	0	0
QoS=bonne	0	0	0	0	0	0
QoS=dégradée	1	1	0	0	0	0
QoS=mauvaise	0	0	1	1	1	1

TABLE 5.27 – Table de probabilités conditionnelles du nœud QoS pour une connectivité mauvaise

connectivité	mauvaise					
	oui			non		
respect du délai	faible	moyenne	forte	faible	moyenne	forte
perte de données = <i>faible</i>	0	0	0	0	0	0
QoS=bonne	0	0	0	0	0	0
QoS=dégradée	0	0	0	0	0	0
QoS=mauvaise	1	1	1	1	1	1

5.5 Scénarios de diagnostic et d'aide à la décision

Pour tester l'efficacité du RB à détecter les défaillances, des scénarios de diagnostic sont présentés en appliquant des observations sur le modèle bayésien modulaire de la figure 5.10.

TABLE 5.28 – Table de probabilités conditionnelles du nœud QoS pour une connectivité perdue

connectivité	perdue					
respect du délai	oui			non		
perte de données	faible	moyenne	forte	faible	moyenne	forte
QoS=bonne	0	0	0	0	0	0
QoS=dégradée	0	0	0	0	0	0
QoS=mauvaise	1	1	1	1	1	1

L'implémentation du RB est faite en utilisant la toolbox Matlab BNT [Mur+01] développée par Kevin Murphy. Les probabilités des nœuds de type *usage* du module bayésien de la partie communication sont initialisées avec des probabilités uniformes. Avant de commencer à exécuter l'application de commande distribuée, la station et le robot échangent des paquets de test (*probe packets*) pour évaluer le délai de transmission et le RSSI. Les procédures d'initialisation et de diagnostic sont illustrées dans la figure 5.14. Des observations certaines sont initialement appliquées sur certains nœuds des couches *usage* et *observation* dans le module bayésien de la partie communication telles que :

- $P(\text{nombre de nœuds communicants} \leq 95) = 1$
- $P(\text{délai} \leq 400ms) = 1$
- $P(\text{distance robot-station} < 30m) = 1$
- $P(\text{obstacle} = \text{oui}) = 1$
- $P(\text{RSSI} > -80dbm) = 1$

Le résultat de l'inférence dans ce module bayésien, après avoir appliqué ces dernières observations certaines, donne les probabilités a posteriori du nœud *QoS* sachant les évidences e : $[P(QoS = \text{bonne}|e); P(QoS = \text{dégradée}|e); P(QoS = \text{mauvaise}|e)] = [0.81; 0.14; 0.05]$. Le nœud *QoS* est un nœud partagé avec le module bayésien de la partie commande. Le lien virtuel qui existe entre les modules bayésiens de la partie communication et de la partie commande permet de définir les probabilités marginales du nœud *QoS* dans la partie commande comme le résultat de l'inférence faite dans la partie communication (voir figure 3.10 du chapitre 3). Les probabilités marginales du nœud *QoS* dans la partie commande sont $[P(QoS = \text{bonne}); P(QoS = \text{dégradée}); P(QoS = \text{mauvaise})] = [0.81; 0.14; 0.05]$. Quant au nœud *état du robot* dans la partie commande, ses probabilités marginales sont définies à partir du résultat de l'inférence dans le module bayésien de la partie opérative. Les probabilités marginales du nœud *état du robot* de la partie commande sont $[P(\text{état du robot} = \text{bon}); P(\text{état du robot} = \text{dégradé}); P(\text{état du robot} = \text{mauvais})] = [0.76; 0.13; 0.11]$.

Les nœuds de la couche *observation* du module bayésien de la partie commande déclenchent la procédure de diagnostic. Tant qu'il n'y a pas de changement dans leurs états, aucune inférence dans ce module ne commencera. L'acquisition de données sur les paramètres observables (couche *observation* et couche *usage*) reste toujours active afin de les utiliser dans

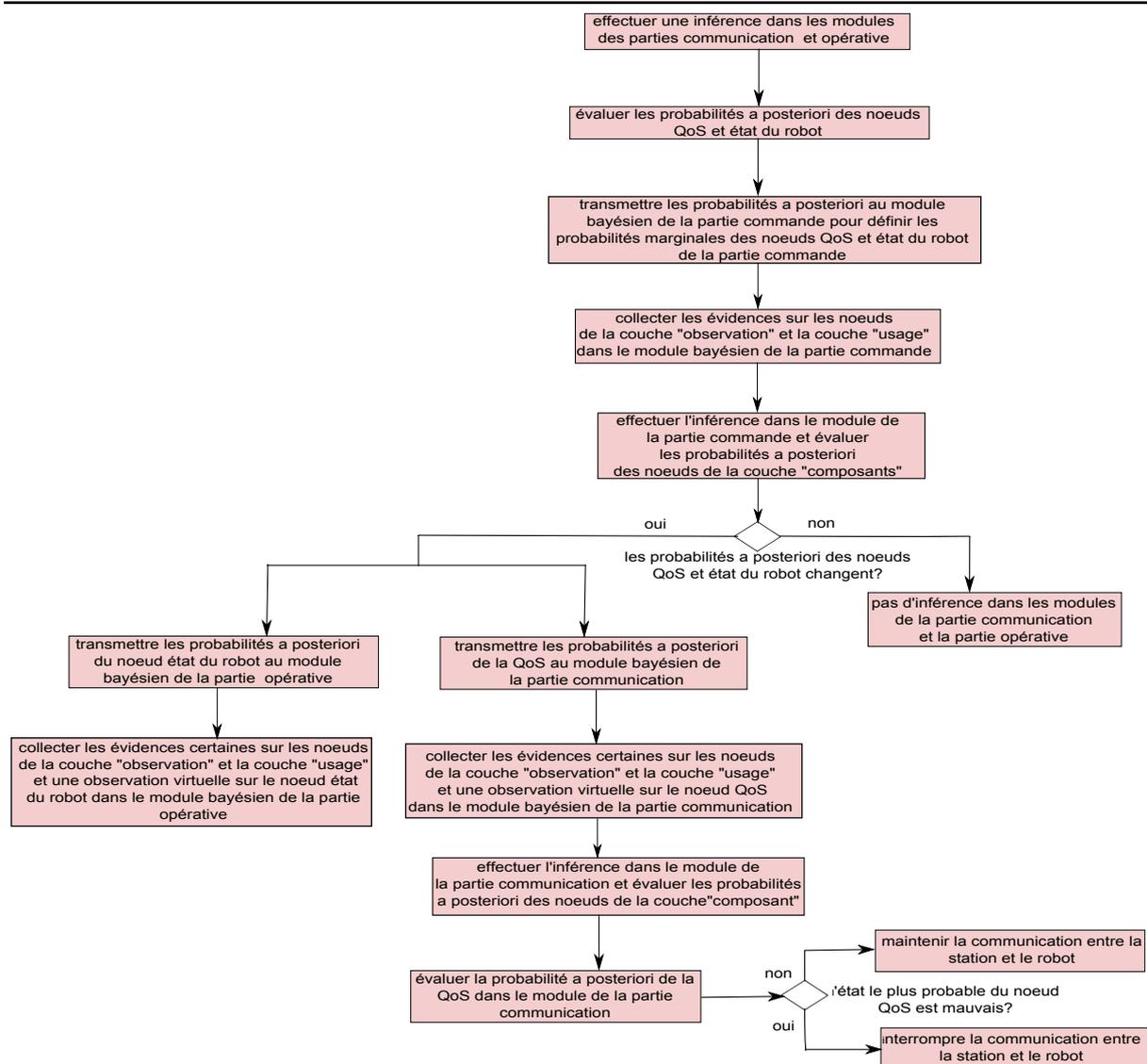


FIGURE 5.14 – Procédures d’initialisation et de diagnostic

la procédure de diagnostic quand elle est exécutée. Dès que l’état d’un des nœuds *distance robot-station*, *stabilité* ou *erreur de position* change, le diagnostic est effectué.

Dans un premier scénario, la fonction de Lyapunov ne vérifie pas la condition de l’inégalité 5.5 et l’erreur de position ne diminue pas au cours du temps, elle est croissante. Deux observations certaines sont alors appliquées :

- $P(\text{stabilité} = \text{non}) = 1$
- $P(\text{erreur de position} = \text{croissante}) = 1$

Une première inférence est alors effectuée au niveau du module bayésien de la partie commande avec un CPU surchargé. Les nœuds de la couche *usage* ont alors les états suivants :

- $P(\text{charge CPU} > s) = 1$

— $P(\text{type de contrôleur} = \text{atteinte de cible}) = 1$

L'inférence est effectuée dans le module de la partie commande et donne les probabilités a posteriori de la figure 5.15. Les probabilités des nœuds *QoS* et *état du robot* ne changent pas par rapport aux valeurs de probabilités marginales a priori de chaque nœud. C'est le calculateur des lois de commandes qui est en mauvais état avec une probabilité de 1. Et dans ce cas, il n'y a pas de deuxième inférence dans les deux autres modules bayésiens.

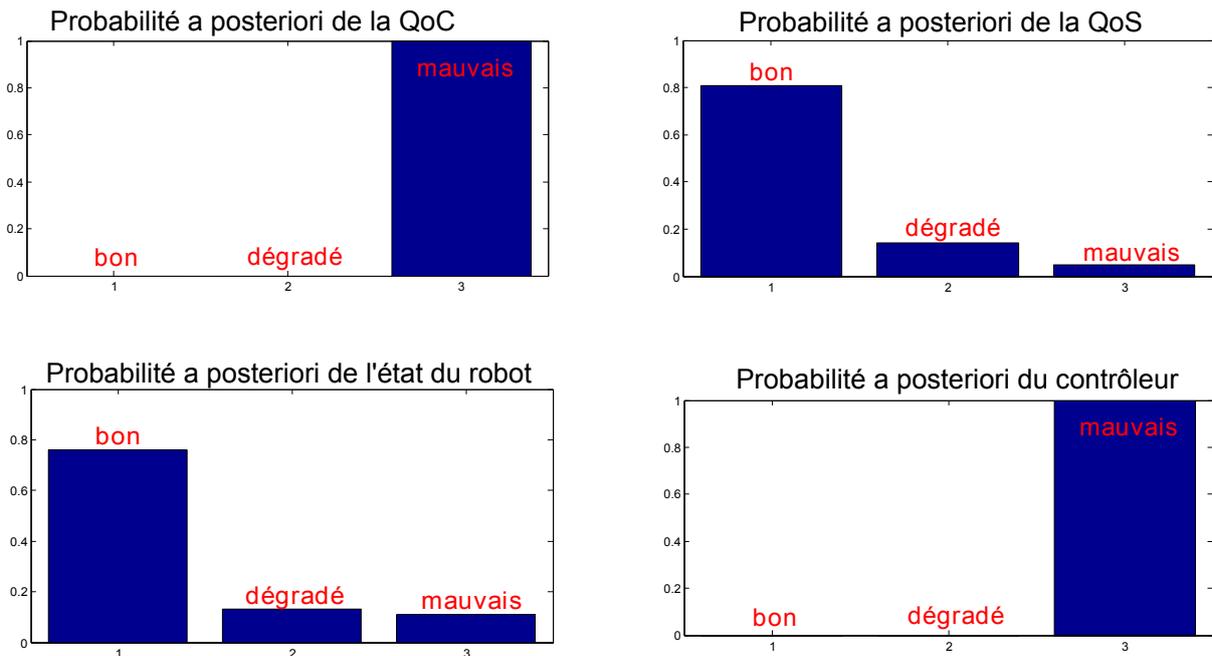


FIGURE 5.15 – Résultats d'inférence : scénario 1

Un deuxième scénario, avec les mêmes observations certaines sur les nœuds *stabilité* et *erreur de position* que sur le premier scénario et un CPU non surchargé, est étudié. Dans ce cas, les probabilités des nœuds de la couche *usage* ont les états suivants :

— $P(\text{charge CPU} < s) = 1$

— $P(\text{type de contrôleur} = \text{atteinte de cible}) = 1$

Les nouvelles probabilités a posteriori des trois nœuds de la couche *composant* dans le module bayésien de la partie commande sont présentées dans la figure 5.16. D'après les résultats de cette première inférence, l'état de la QoC est mauvais avec une probabilité de 1, un résultat conforme à ce que nous avons défini pour la QoC selon les états de l'erreur de position et la stabilité dans l'algorithme 1. Le calculateur des lois de commandes (ou contrôleur) est en bon état avec une probabilité de 0.71, ce qui est bien le cas vu que le CPU n'est pas surchargé (voir table 5.4). Les probabilités a posteriori de la QoS, qui sont $[0.35; 0.09; 0.56]$, montrent que la cause de la dégradation de la QoC est le mauvais état de la QoS avec une probabilité de 0.56. En outre, l'état du nœud *état du robot* reste bon avec une probabilité de 0.48, dégradé avec une valeur de 0.15 et mauvais avec une pro-

tabilité de 0.37. Les valeurs des probabilités a posteriori des nœuds *QoS* et *état du robot*

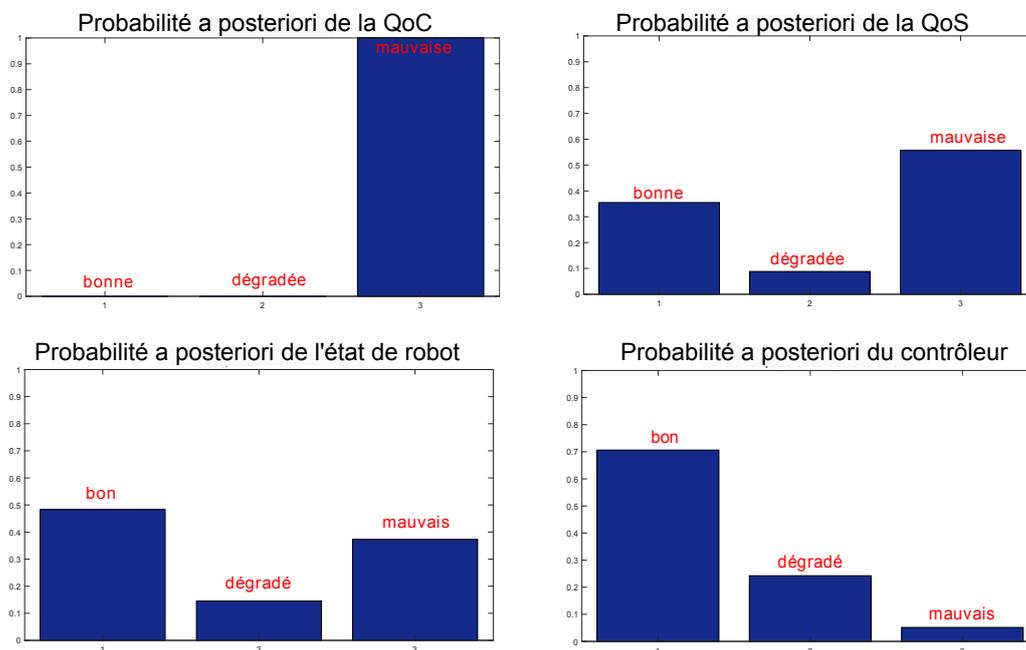


FIGURE 5.16 – Résultats d'inférence : scénario 2

sont transmises, via le lien virtuel, vers les mêmes nœuds dans les modules bayésiens de la partie communication et de la partie opérative. Deux autres inférences sont réalisées.

Dans le module de la partie opérative, deux observations certaines sont appliquées sur les nœuds $P_{erreur,gauche}$ et $P_{erreur,droit}$ telles que $P(P_{erreur,gauche} \leq 10\%) = 1$ et $P(P_{erreur,droit} \leq 10\%) = 1$ et une observation incertaine est appliquée sur le nœud *état du robot* telle que $P[\text{état du robot} = \text{bon}; \text{état du robot} = \text{dégradé}; \text{état du robot} = \text{mauvais}] = [0.48; 0.15; 0.37]$. Les deux moteurs et le robot sont en bon état.

Dans le module bayésien de la partie communication, trois observations certaines sont appliquées sur les nœuds *taux de pertes de paquets*, *délat* et *débit* avec les observations certaines initialement définies sur les nœuds de type *usage*. Une observation incertaine/virtuelle sur le nœud *QoS* du module bayésien de la partie communication est aussi appliquée. Nous avons :

- observation incertaine :
 - $P[\text{QoS}=\text{bonne}, \text{QoS}=\text{dégradée}, \text{QoS}=\text{mauvaise}] = [0.35; 0.09; 0.56]$
 - $P(90\% \leq \text{taux de pertes de paquets} \leq 100\%) = 1$
 - $P(\text{délai} > 400\text{ms}) = 1$
 - $P(\text{débit} < 3.2\text{kbit/s}) = 1$

Le résultat de l'inférence, dans le module bayésien de la partie communication, donne les probabilités a posteriori suivantes :

- $P(\text{QoS} = \text{mauvaise} | e') = 1$

- $P(\text{connectivité} = \text{bonne} | e') = 1$
- $P(\text{respect du délai} = \text{non}) = 1$
- $P(\text{canal de communication} = \text{saturé}) = 1$
- $P(\text{perte de données} = \text{forte}) = 1$

avec e' l'ensemble des observations dans le module bayésien de la partie communication.

Les probabilités a posteriori des nœuds de la couche *composant* montrent qu'il n'y a pas de problème de connectivité, la QoS est mauvaise à cause des pertes, des délais et de la saturation du canal. Nous évaluons les états des nœuds de la couche *usage*, l'état du nœud *charge réseau* est tel que $P(\text{charge réseau} \geq 6\text{Mbit/s}) = 0.53$. Par conséquent, une charge réseau, autre que celle de l'application de commande, cause la dégradation de l'état du réseau, il s'agit d'une charge due à une application gourmande ou d'une cyberattaque. Quand le réseau ne fournit pas le niveau requis de la QoS, une stratégie de rétablissement de bon comportement doit être mise en œuvre. Quand la QoS est mauvaise, le robot commute du fonctionnement débarqué au fonctionnement embarqué, il devient autonome et il ne reçoit plus les commandes de la station. Pour prendre la décision de rompre la connexion entre la station et le robot, l'algorithme 2 est implémenté sur la station. Par la suite, la station de contrôle envoie l'ordre de déconnexion, un message d'alarme prioritaire, au robot. Le robot commute alors sur un fonctionnement embarqué. La question qui se pose maintenant est la suivante : **comment l'architecture de diagnostic évolue-t-elle après ce changement de mode de fonctionnement ?** La réponse à cette question est l'objectif de la section 5.6.

Algorithm 2 Algorithme de prise de décision

```

1:  $P_1 \leftarrow P(QoC = \text{bonne} | \text{observations})$ 
2:  $P_2 \leftarrow P(QoC = \text{dégradée} | \text{observations})$ 
    $P_3 \leftarrow P(QoC = \text{mauvaise} | \text{observations})$ 
4:  $P'_1 \leftarrow P(QoS = \text{bonne} | \text{observations})$ 
    $P'_2 \leftarrow P(QoS = \text{dégradée} | \text{observations})$ 
6:  $P'_3 \leftarrow P(QoS = \text{mauvaise} | \text{observations})$ 
   if [ $P_1 > \max(P_2, P_3)$ ] || [ $P_2 > \max(P_1, P_3)$ ] then
8:    $order \leftarrow 0$  //le robot reste connecté
   else
10:  if  $P'_3 > \max(P'_2, P'_1)$  then
        $order \leftarrow 1$  //le robot doit se déconnecter
12:  end if
   end if

```

5.6 Reconfiguration de l'architecture de diagnostic

Il existe trois modules bayésiens comme cela a été détaillé dans la section 5.3 : le module de la partie commande, le module de la partie communication et le module de la partie opérative. Quand la station prend la décision d'arrêter la communication avec le robot en utilisant l'algorithme 2 de prise de décision, le robot devient autonome jusqu'à ce qu'il trouve une nouvelle station de contrôle. Cette prise de décision est effectuée si la QoS devient mauvaise à cause d'un défaut réseau qui a causé une dégradation de la QoS. La nouvelle station (le nouveau réseau) de communication doit garantir le niveau de QoS requis.

Après avoir commuté vers le contrôleur embarqué, le système change alors d'architecture et de mode de fonctionnement comme cela a été décrit dans la partie 5.2.1. Le réseau de communication ne fait plus partie du système, son état et sa QoS n'influencent plus la performance du système. Le module bayésien de la partie communication n'intervient plus dans la procédure de diagnostic. Le nœud partagé QoS doit être enlevé de l'ensemble des parents du nœud QoC puisque la dépendance cause-à-effet n'existe plus comme le montre la figure 5.17. Nous devons alors mettre à jour l'architecture de diagnostic : une nouvelle architecture qui ne prend en compte que la partie commande et la partie opérative pour surveiller l'avancement de la mission du robot. Dans cette nouvelle configuration, le diagnostic consiste en un RB monolithique qui est le résultat de la combinaison des modules bayésiens de la partie opérative et de la partie commande. Quand le robot se connecte à une nouvelle station de contrôle, le système reprend son fonctionnement débarqué, le module bayésien de la partie communication ré-intègre la procédure de diagnostic : un diagnostic distribué, et le nœud *QoS* est rajouté à l'ensemble de parents du nœud *QoC*. Les procédures de commutations entre les modes de diagnostic vont être détaillées dans les deux sous-sections 5.6.1 et 5.6.2.

5.6.1 Commutation du diagnostic distribué au diagnostic embarqué

Dans le cas d'une dégradation de la QoC à cause de la QoS, le robot commute sur son contrôleur embarqué s'il ne trouve pas une nouvelle station de contrôle pouvant assurer la continuité du service. Le contrôleur embarqué est implémenté sur le robot qui devient autonome, et le réseau de communication ne constitue plus un composant du système global. L'architecture de diagnostic est alors mise à jour pour s'adapter à la nouvelle configuration du système comme cela a été décrit précédemment. Une première étape de cette commutation consiste à supprimer le module bayésien de la partie communication et à enlever le nœud *QoS* de l'ensemble des parents du nœud *QoC* dans le module bayésien de commande. Les probabilités conditionnelles de la *QoC* doivent être redéfinies, puisque

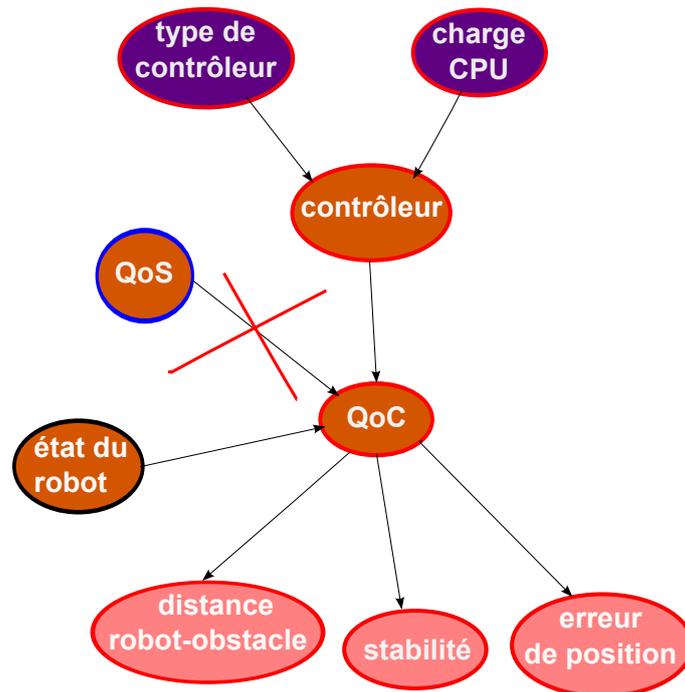


FIGURE 5.17 – Suppression du nœud QoS

l'état du nœud QoC ne dépend plus que de l'état des nœuds *contrôleur* et *état robot*, en utilisant l'équation 5.6 :

$$P(QoC|R, C) = \frac{\sum_{QoS} P(QoC|R, C, QoS)}{\sum_{QoC} P(QoC|R, C)} \quad (5.6)$$

avec R : le nœud état du robot et C : le nœud contrôleur.

Une deuxième étape consiste à commuter sur le diagnostic embarqué en transformant la structure bayésienne distribuée en une structure monolithique embarquée sur le robot. Cela revient à réunir les modules bayésiens de la partie commande et la partie opérative dans un seul RB monolithique. Une procédure de validation pour le RB assemblé doit être effectuée pour vérifier que la structure bayésienne obtenue supporte une inférence cohérente et concise. Cette procédure est inspirée du travail de [Oud+10] qui vérifie que l'inférence globale est correcte dans sa structure modulaire. La méthode décrite dans [Oud+10] n'est pas utilisée pour construire le RB modulaire dans notre travail mais pour rassembler les modules bayésiens dans une structure monolithique. La procédure de construction que nous avons utilisée, définie dans [LYC15], garantit que le RB monolithique correspondant aux modules bayésiens est un graphe acyclique avec une inférence modulaire indépendante. La procédure décrite dans [Oud+10] assure une inférence globale correcte en construisant un graphe d'inférence pour les modules bayésiens. La procédure d'assemblage de [Oud+10] rajoute les contraintes et les dépendances que [LYC15] supprime. Cette procédure est appliquée seulement sur les modules de la partie opérative et la partie commande.

Deux modules bayésiens peuvent être rassemblés dans une seule structure monolithique s'ils forment un graphe d'inférence qui garantit un passage de messages cohérent dans l'algorithme d'inférence. Soit $C = \{C_1, C_2, \dots, C_n\}$ l'ensemble des cliques (*clusters*) tel que chaque clique correspond à un module bayésien du RB modulaire et n est le nombre de modules bayésiens. Soit $S = \{S_{i,j} | 1 \leq i \leq n, 1 \leq j \leq n\}$ l'ensemble de séparateurs entre les cliques tel que $S_{i,j} = C_i \cap C_j \neq \emptyset$. Deux modules bayésiens ψ_i et ψ_j , dont les ensembles de variables sont V_i et V_j respectivement, échangent des probabilités marginales via leurs nœuds communs ($V_i \cap V_j$) dans une structure bayésienne modulaire. Le graphe d'inférence $G_{\text{inférence}} = (C, S)$ est, par définition, un graphe qui contient l'ensemble des cliques et tous les séparateurs qui contiennent les nœuds partagés échangeant les probabilités marginales a posteriori [Oud+10]. Une étape très importante de cette procédure d'assemblage consiste à trouver l'ensemble des cycles Σ dans le RB monolithique obtenu suite à l'opération d'assemblage des RBs modulaires. Après avoir obtenu Σ , il faut éliminer cet ensemble de cycles pour construire un graphe de cliques acycliques capable de supporter une inférence globale correcte. [Oud+10] ont défini deux méthodes d'élimination de cycles qui sont adoptées dans notre travail, et ont prouvé la cohérence de l'inférence globale sur le nouveau graphe obtenu.

- **Élimination de séparateurs** : si un ensemble de cliques crée un cycle à cause d'un séparateur redondant ou d'une variable redondante (cycle dégénéré), il faut éliminer le plus petit séparateur qui contient la variable redondante dans tous les séparateurs du cycle [XL03]. Le plus petit séparateur contenant la variable redondante peut être composé d'une seule variable partagée entre les cliques du cycle. Le graphe résultant de cette opération est un arbre de cliques.
- **Instanciation d'une variable** : si le cycle n'est pas dû à un séparateur redondant (cycle non dégénéré), il faut instancier une variable X qui appartient à au moins un séparateur entre deux cliques du cycle. [XL03] a prouvé que l'inférence n'est pas cohérente dans un graphe de cliques avec ce type de cycle. En effet, pour deux distributions de probabilités initiales différentes avec l'instanciation de la même variables dans les deux cas, il a obtenu les mêmes probabilités a posteriori. Les probabilités a posteriori ne sont plus sensibles au changement des probabilités a priori. Le nouveau graphe d'inférence obtenu est le résultat de l'élimination de la variable X de tous les séparateurs et des cliques du graphe d'inférence contenant le cycle, c'est un graphe structuré en arbre. Le séparateur correspondant à cette variable supprimée n'existe plus et le cycle est alors éliminé.

Après avoir éliminé les cycles pour obtenir un graphe d'inférence cohérent, les nœuds partagés sont appelés les *d-sep nœuds*. En effet, les parents de chaque nœud sont définis dans un seul module bayésien et par conséquent les distributions de probabilités conditionnelles définies dans le RB modulaire et le RB monolithiques sont les mêmes. Le RB monolithique correspondant à la structure modulaire est alors validé pour une inférence correcte.

La procédure d'assemblage est utilisée pour rassembler les modules de la partie opérative et

la partie commande dans une structure bayésienne unique comme le montre la figure 5.18. Les parents du nœud *état du robot* sont tous définis dans un seul module et par conséquent ils correspondent à une seule clique dans le RB monolithique. Le module bayésien de la partie commande est la deuxième clique et le nœud partagé *état du robot* est le *d-sepnode* dans le RB monolithique. Cette procédure d'assemblage permet d'obtenir les deux cliques :

$$C_1 = \{batterie, \text{moteur gauche}, \text{moteur droit}, \text{erreur de consigne gauche}, \text{erreur de consigne droite}, \text{état de capteurs}, \text{coefficient de performance}, \text{état du robot}\}$$

$$C_2 = \{\text{type de contrôleur}, \text{charge CPU}, \text{contrôleur}, \text{distance robot-obstacle}, \text{stabilité}, \text{erreur de position}, \text{QoC}\}$$

et le séparateur $S = \{\text{état du robot}\}$.

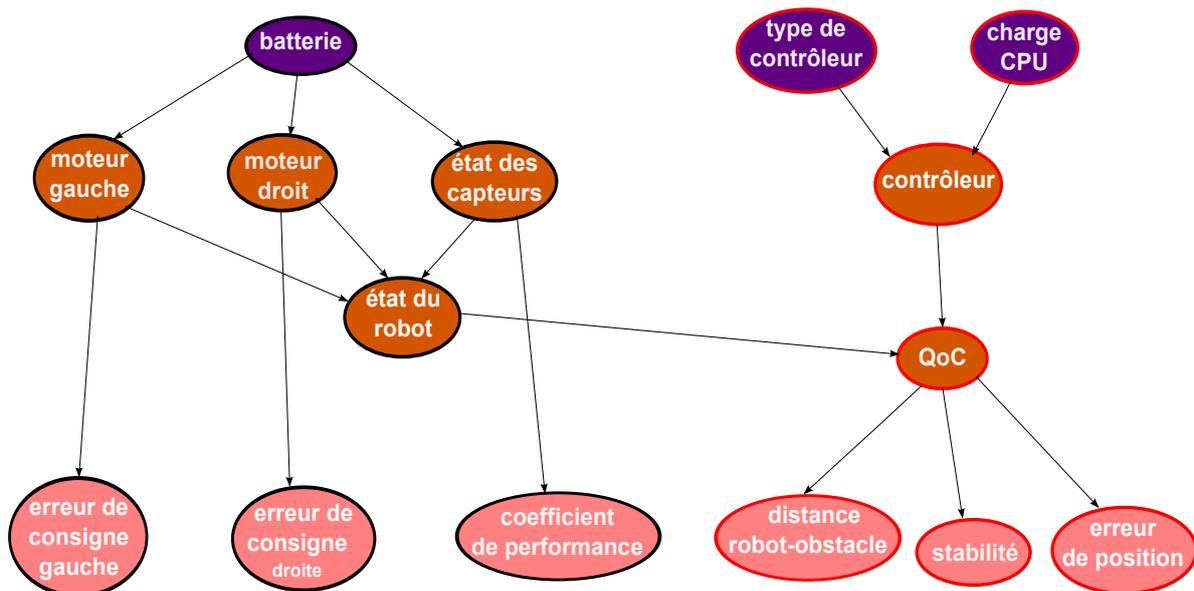


FIGURE 5.18 – Réseau Bayésien monolithique embarqué

Il n'existe pas de cycles dans le graphe de cliques correspondant au RB de la figure 5.18 et par conséquent il n'y a pas besoin des deux procédures d'élimination de cycles. Cette architecture de diagnostic embarqué est adoptée tant que le robot est autonome et jusqu'à ce qu'il trouve une nouvelle station de contrôle qui déclenchera le rétablissement d'un diagnostic distribué et des RBs modulaires.

5.6.2 Commutation du diagnostic embarqué au diagnostic distribué

Une fois que le robot trouve une nouvelle station de contrôle (un nouveau contrôleur distant), l'architecture de diagnostic embarqué sur le robot n'est plus nécessaire. Le retour au diagnostic distribué modulaire est primordial, c'est-à-dire avoir les trois modules bayésiens de diagnostic tels que la partie opérative reste en embarqué, et les parties communication et commande sont à nouveau implémentées sur la station de contrôle. La première étape de commutation vers le diagnostic distribué consiste à modulariser le RB monolithique de la figure 5.18 en se basant sur la procédure de modularisation décrite dans [HC09] qui sert à définir les variables suivant les domaines des modules. Le premier domaine correspond à la partie opérative : ses composants et leurs observations. Les variables du premier RB sont donc :

$$M_1 = \{ \text{batterie, moteur gauche, moteur droit, erreur de consigne gauche, erreur de consigne droite, état des capteurs, coefficient de performance, état du robot} \}$$

Le deuxième domaine est celui de la partie commande. Les variables du deuxième module bayésien sont donc :

$$M_2 = \{ \text{type de contrôleur, charge CPU, contrôleur, distance robot-obstacle, stabilité, erreur de position, QoS} \}$$

Les variables conservent leurs distributions de probabilités dans les RBs locaux. Un apprentissage des paramètres en ligne va ensuite mettre à jour les probabilités des variables. Après la modularisation, la probabilité marginale du nœud commun *état du robot* est calculée grâce à l'application de l'inférence sur le module de la partie opérative. Cette probabilité marginale obtenue va ensuite être attribuée au nœud racine *état du robot* dans le module bayésien de la partie commande. Le module bayésien de communication est intégré avec des probabilités a priori apprises à partir de la base de données implémentée sur la station de contrôle. Les probabilités conditionnelles sont définies à partir des communications déjà existantes entre d'autres robots et la station de contrôle. Si les probabilités conditionnelles ne sont pas disponibles, un apprentissage en ligne doit être réalisé.

Pour réintégrer le nœud *QoS* dans l'ensemble des parents du nœud *QoS*, nous avons établi une procédure de combinaison de RBs inspirée de la procédure de consensus de [HW13]. Le consensus des RBs est le résultat de la combinaison de plusieurs RBs dont l'apprentissage est effectué à partir des bases de données ou des avis des experts. La procédure de combinaison dans le consensus des RBs consiste à combiner les graphes correspondants à ces RBs et à agréger les distributions de probabilités [DM03]. Le travail de [HW13] propose une méthode de combinaison en deux étapes : une combinaison qualitative pour combiner les modèles graphiques et une combinaison quantitative pour obtenir les distributions de

probabilités. La procédure de consensus est généralement appliquée sur des RBs ayant les mêmes variables. Cependant, [HW13] ont proposé une méthode de combinaison de RBs définis sur des ensembles de variables différents, ce qui est le cas de notre problématique.

5.6.2.1 Combinaison qualitative et quantitative des réseaux bayésiens

Pour résoudre le problème de reconfiguration de l'architecture de diagnostic, l'étape d'extension dans l'algorithme de combinaison et d'agrégation de [HW13] est utilisée. Soient RB_1 et RB_2 les deux RBs qui vont subir la procédure de combinaison telle que RB_1 est étendu en $RB_1 \oplus RB_2$, et V_1 et V_2 leurs ensembles de variables respectifs qui vérifient les conditions suivantes :

- $V_1 \neq V_2$
- $V_1 \cap V_2 \neq \emptyset$

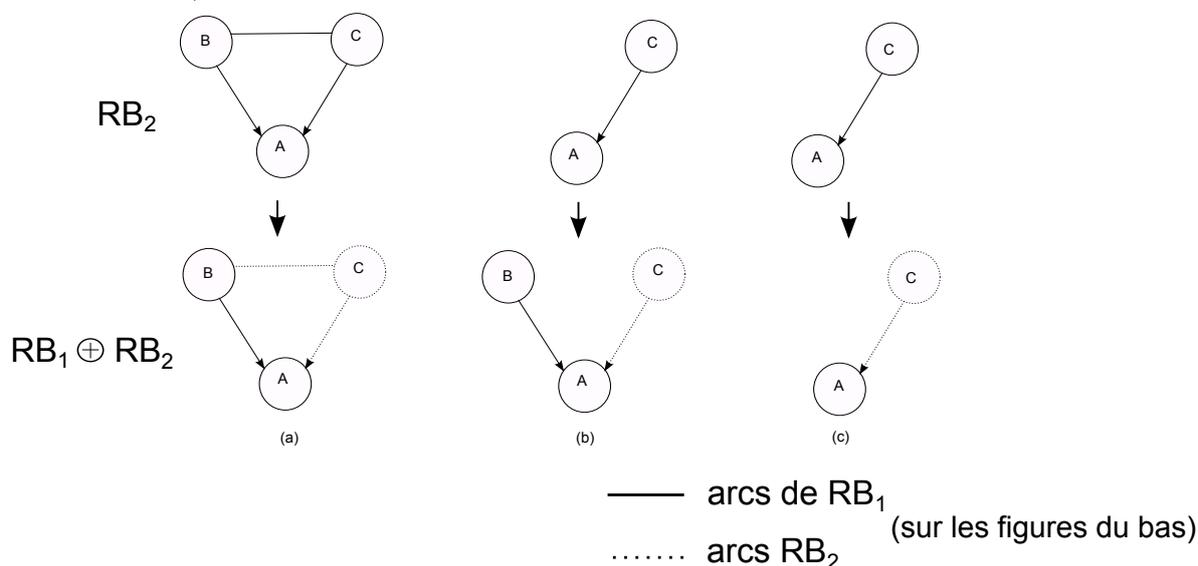


FIGURE 5.19 – Combinaison qualitative des réseaux bayésiens [HW13]

La procédure d'extension des RBs consiste en :

- **une extension qualitative** : soient G_1 et G_2 les deux graphes acycliques correspondant à RB_1 et RB_2 respectivement. G_1 est étendu en $G_1 \oplus G_2$ tel que $G_1 \oplus G_2 = \{V_1 \cup V_2, E_1 \cup E\}$, où E est l'ensemble des arcs $\{(C, A) \in E_2 | C \in V_2 - V_1, A \in V_1\}$. Les lignes continues dans la figure 5.19 représentent les arcs de RB_1 et les lignes en pointillé sont les arcs ajoutés quand RB_1 est étendu en $RB_1 \oplus RB_2$. Les variables dans le graphe résultant de la procédure de l'extension doivent être consistantes, c'est-à-dire que le graphe $G_1 \oplus G_2$ doit être acyclique.
- **une extension quantitative** : cette étape consiste à calculer les tables de probabilités conditionnelles de chaque nœud de $G_1 \oplus G_2$. Si $A \in V_2 - V_1$ ou $A \in V_1 - V_2$, alors

la table de probabilités conditionnelles de A ne change pas. Si $A \in V_1$ et $C \in V_2 - V_1$ tels que $(C, A) \in E_2$, la table de probabilités conditionnelles de A est obtenue (cas (a) de la figure 5.19) en utilisant l'équation 5.7 avec $A, B \in V_1, C \in V_2 - V_1$.

$$P(A = a_i | B = b_j, C = c_k) = \frac{P_1(A = a_i | B = b_j) \times \frac{P_2(C=c_k | A=a_i, B=b_j)}{P_2(C=c_k | B=b_j)}}{\sum_{i=1}^n (P_1(A = a_i | B = b_j) \times \frac{P_2(C=c_k | A=a_i, B=b_j)}{P_2(C=c_k | B=b_j)}} \quad (5.7)$$

avec P_1 et P_2 les distributions de probabilités dans RB_1 et RB_2 respectivement. Pour le cas (b) de la figure 5.19, la variable B est indépendante de C car aucune dépendance entre B et C n'est définie dans RB_2 . Les probabilités conditionnelles de A sont calculées selon l'équation 5.8 :

$$P(A = a_i | B = b_j, C = c_k) = \frac{P_1(A = a_i | B = b_j) \times \frac{P_2(C=c_k | A=a_i)}{P_2(C=c_k)}}{\sum_{i=1}^n (P_1(A = a_i | B = b_j) \times \frac{P_2(C=c_k | A=a_i)}{P_2(C=c_k)})} \quad (5.8)$$

avec $\frac{P_2(C=c_k | A=a_i)}{P_2(C=c_k)} = \frac{P_2(A=a_i | C=c_k)}{P_2(A=a_i)}$.

Pour calculer les probabilités conditionnelles de A dans le cas (c) de la figure 5.19 où la variable A n'est pas connectée à une autre variable B comme dans les cas (a) et (b), l'équation 5.9 est utilisée :

$$P(A = a_i | C = c_k) = \frac{P_1(A = a_i) \times \frac{P_2(C=c_k | A=a_i)}{P_2(C=c_k)}}{\sum_{i=1}^n (P_1(A = a_i) \times \frac{P_2(C=c_k | A=a_i)}{P_2(C=c_k)})} \quad (5.9)$$

5.6.2.2 Procédure d'extension du module bayésien de la partie commande

Le module bayésien de la partie commande sans le nœud QoS de la figure 5.17 bénéficie de la procédure d'extension pour s'adapter à l'évolution de l'architecture du système. Quand le réseau de communication réintègre la boucle de commande du robot, le nœud QoS est ajouté à l'ensemble des parents du nœud QoC . Le module bayésien de la partie commande RB_1 de la figure 5.20, obtenu après la modularisation du RB monolithique, est combiné à RB_2 . RB_2 est composé de la dépendance causale entre les nœuds QoS et QoC comme le montre la figure 5.20. Nous utilisons la procédure d'extension et de combinaison décrite dans la sous section 5.6.2.1. La table de probabilités conditionnelles du nœud QoC est mise à jour en utilisant l'équation 5.8. L'extension quantitative décrite dans l'équation 5.10 correspond au cas pour lequel nous avons état du robot = *bon*, contrôleur = *bon*,

QoS = *bonne* et QoC = *dégradée*.

$$P(QoC = \text{dé} | R = b, C = b, QoS = b) = \frac{P_1(QoC = \text{dé} | R = b, C = b) \times \frac{P_2(QoC = \text{dé} | QoS = b)}{P_2(QoC = \text{dé})}}{\sum_{s=b, \text{dé}, m} P_1(QoC = s | R = b, C = b) \times \frac{P_2(QoC = s | QoS = b)}{P_2(QoC = s)}} \quad (5.10)$$

avec R : nœud état du robot, C : nœud contrôleur, b : bon/bonne, dé : dégradée. La probabilité marginale du nœud QoS est calculée par l'algorithme d'inférence dans le module bayésien de la partie communication.

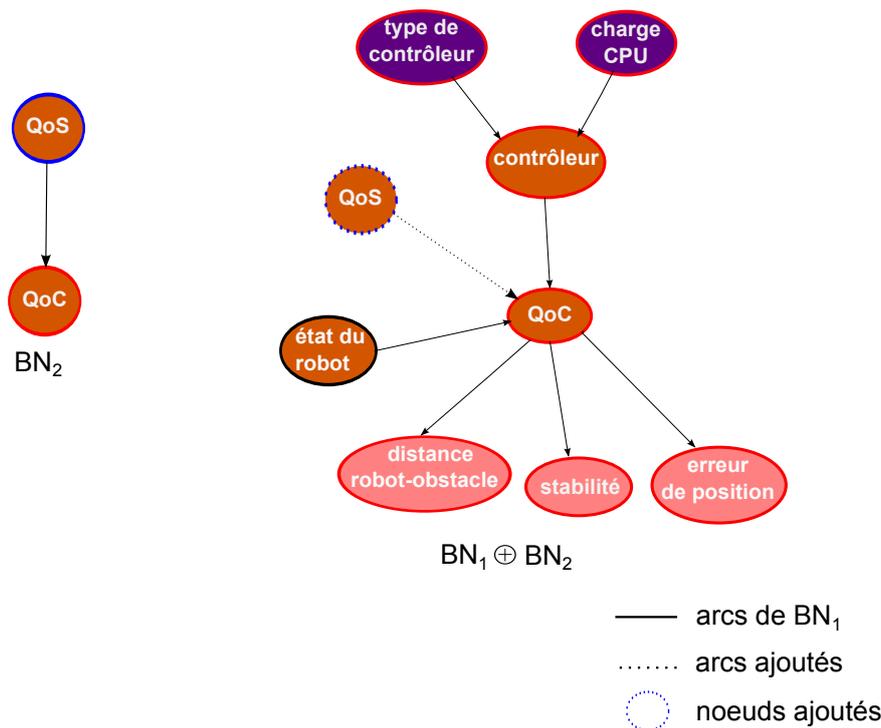


FIGURE 5.20 – Extension du module bayésien de commande

5.6.3 Scénario de diagnostic et de commutation

Nous appliquons la méthode de diagnostic et les méthodes de commutation sur le robot simulé. La mission du robot consiste à atteindre sa destination (30, 20) en partant de la position initiale (0, 0). Nous disposons de deux stations de contrôle, la première a pour coordonnées (1, 1) et la deuxième se situe à la position (25, 25). Au début, le robot reçoit les commandes de la première station de contrôle. L'état du réseau, dans le premier réseau de communication, où la première station est le point d'accès, est mauvais avec un taux de pertes de paquets de 95%. Ces mauvaises conditions provoquent la dégradation de

la QoC comme nous l'avons montré sur la figure 5.6. L'objectif est alors d'améliorer le comportement du réseau. Nous prenons en compte les mêmes observations que celles du deuxième scénario de la section 5.5 sur les nœuds *stabilité* et *erreur de position*. Cette dégradation et ces observations sont détectées quand le robot est à la position $(X; Y) = (0.68; 1.71)$ comme le montre la figure 5.21. Avant cette position, le robot n'évolue pas

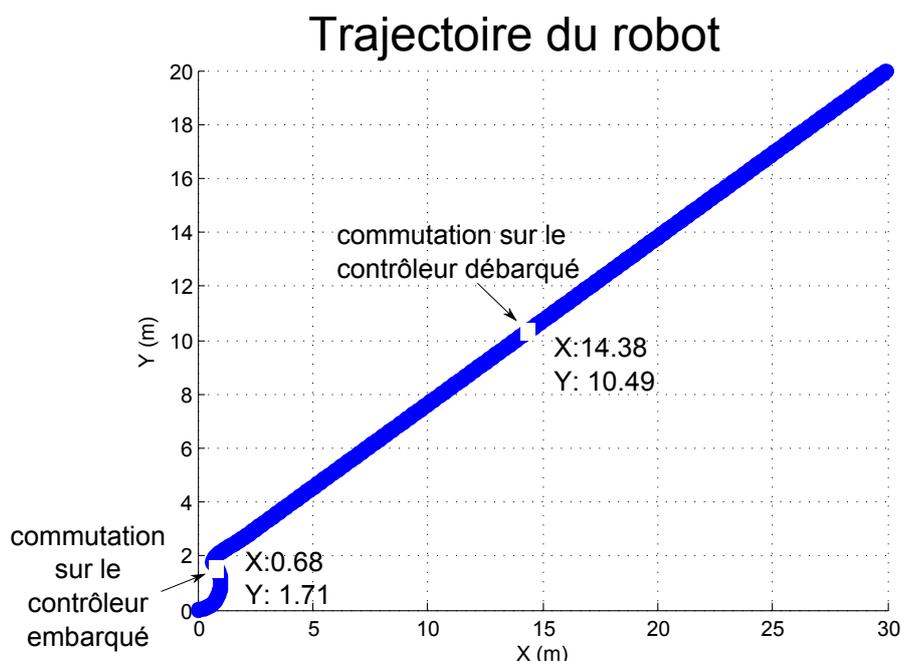


FIGURE 5.21 – Trajectoire du robot

vers sa position finale. Le début de son comportement est similaire à celui de la figure 5.6. Cette déviation du comportement attendu est détectée parce que l'erreur de position ne diminue pas et que la stabilité n'est pas garantie puisque la fonction de Lyapunov n'est pas décroissante. Après les deux inférences faites au niveau des modules bayésiens, l'algorithme 2 de la section 5.5 est alors utilisé pour prendre la décision de commutation sur le contrôleur embarqué vu que le robot n'est pas encore dans la zone de couverture de la deuxième station (zone de couverture de rayon de 18m). Les conditions de commutation sont satisfaites :

- dans le réseau bayésien de la partie commande : $P(QoC = m|e) > P(QoC = dé|e)$
& $P(QoC = m|e) > P(QoC = b|e)$
avec $P(QoC = m|e) = 1$, $P(QoC = dé|e) = 0$ et $P(QoC = b|e) = 0$
- dans le réseau bayésien de la partie communication : $P(QoS = m|e) > P(QoS = dé|e)$
& $P(QoS = m|e) > P(QoS = b|e)$
avec $P(QoS = m|e) = 1$, $P(QoS = dé|e) = 0$ et
 $P(QoS = b|e) = 0$

Au moment de la commutation, l'architecture de diagnostic est alors mise à jour et elle est

embarquée sur le robot comme cela a été détaillé dans la sous section 5.6.1. La première étape consiste à supprimer le nœud QoS des tables de probabilités conditionnelles 5.5, 5.6 et 5.7 du nœud QoC en utilisant l'équation 5.6. La nouvelle table de probabilités du nœud QoC est représentée dans les tables 5.29, 5.30 et 5.31. Nous avons vérifié dans la sous

TABLE 5.29 – Table de probabilités conditionnelles de l'état de la QoC sachant que l'état du contrôleur est bon

contrôleur	bon		
	bon	dégradé	mauvais
QoC=bonne	0.54	0.53	0.37
QoC=dégradée	0.18	0.10	0.29
QoC=mauvaise	0.28	0.37	0.34

TABLE 5.30 – Table de probabilités conditionnelles de l'état de la QoC sachant un état dégradé du contrôleur

contrôleur	dégradé		
	bon	dégradé	mauvais
QoC=bonne	0	0	0
QoC=dégradée	0.66	0.66	0
QoC=mauvaise	0.34	0.34	1

TABLE 5.31 – Table de probabilités conditionnelles de l'état de la QoC sachant un état mauvais du contrôleur

contrôleur	mauvais		
	bon	dégradé	mauvais
QoC=bonne	0	0	0
QoC=dégradée	0	0	0
QoC=mauvaise	1	1	1

section 5.6.1 que l'assemblage des deux modules bayésiens de la partie opérative et de la partie commande ne forme pas un graphe cyclique. Le réseau bayésien monolithique de la figure 5.18 est utilisé comme outil de diagnostic embarqué. Le réseau bayésien embarqué continue la supervision de l'état de la QoC et de l'état du robot. Un apprentissage des paramètres est disponible et est effectué car les performances du fonctionnement embarqué ne sont pas comme celles du fonctionnement débarqué. Les probabilités conditionnelles du nœud QoC des tables 5.29, 5.30 et 5.31 sont alors améliorées par un apprentissage des paramètres comme le montrent les tables de probabilités conditionnelles 5.32, 5.33 et 5.34.

Les tables de probabilités conditionnelles obtenues par apprentissage sont plus pertinentes que celles du calcul mais elles ne sont pas toujours disponibles. Le robot continue à calculer

TABLE 5.32 – Table de probabilités conditionnelles par apprentissage de l'état de la QoC sachant un bon état du contrôleur

contrôleur	bon		
état du robot	bon	dégradé	mauvais
QoC=bonne	0.93	0.89	0
QoC=dégradée	0.064	0.10	0.95
QoC=mauvaise	0.006	0.01	0.05

TABLE 5.33 – Table de probabilités conditionnelles par apprentissage de l'état de la QoC sachant un état dégradé du contrôleur

contrôleur	dégradé		
état du robot	bon	dégradé	mauvais
QoC=bonne	0.87	0.87	0.54
QoC=dégradée	0.11	0.11	0.34
QoC=mauvaise	0.02	0.02	0.12

TABLE 5.34 – Table de probabilités conditionnelles par apprentissage de l'état de la QoC sachant un état mauvais du contrôleur

contrôleur	mauvais		
état du robot	bon	dégradé	mauvais
QoC=bonne	0	0	0
QoC=dégradée	0	0	0
QoC=mauvaise	1	1	1

les estimations statistiques et les observations (erreur de position, stabilité, distance robot-station). Une observation est faite sur l'état du moteur gauche qui devient dégradé tel que $P(10\% < P_{erreur,gauche} \leq 30\%) = 1$. L'état du nœud *QoC* reste bon. Les probabilités a posteriori de l'état du nœud *état du robot* sont calculées :

- $P(\text{état du robot} = \text{bon}|e) = 0.49$
- $P(\text{état du robot} = \text{dégradé}|e) = 0.49$
- $P(\text{état du robot} = \text{mauvais}|e) = 0.02$

Cette probabilité a posteriori représente la probabilité marginale de l'état du nœud *état du robot* quand le robot détecte la deuxième station. D'autres observations sont importantes à transmettre au diagnostic débarqué à savoir les observations sur les nœuds *stabilité* et

erreur de position. Le robot commute vers le contrôleur débarqué à l'instant 81.7s. La procédure de commutation garantit la continuité de la stabilité et du bon comportement du robot comme le montre la figure 5.22. Quand le robot se connecte à la nouvelle station

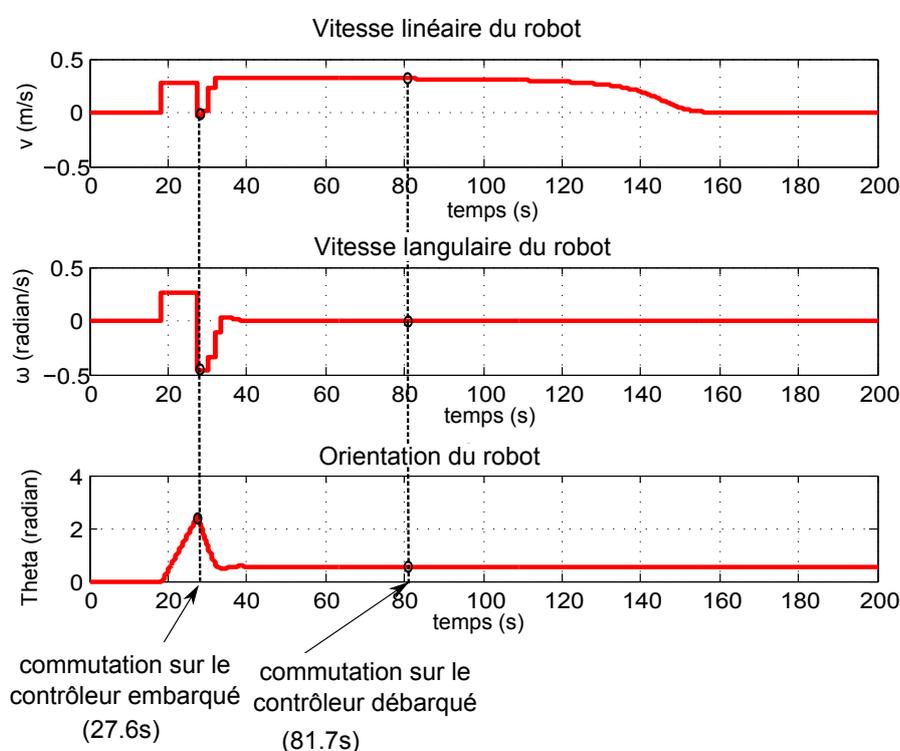


FIGURE 5.22 – Vitesses et orientation du robot pendant sa mission de navigation

de contrôle, le nœud QoS doit être réintégré à la table de probabilités conditionnelles du nœud QoC et le module bayésien de la partie communication est de nouveau utilisé. La dépendance de cause-à-effet entre la QoS et la QoC est quantifiée dans le nouveau réseau de communication. Elle est définie dans la base de données de la nouvelle station de contrôle qui enregistre les données des applications de commande qui partagent le réseau. Les probabilités conditionnelles de cette dépendance sont définies dans la table 5.35. La

TABLE 5.35 – Table de probabilités conditionnelles de la dépendance de cause-à-effet entre la QoS et la QoC

QoS	bonne	dégradée	mauvaise
QoC =bonne	0.26	0.23	0
QoC =dégradée	0.29	0.31	0.04
QoC =mauvaise	0.46	0.46	0.96

dépendance entre la QoS et la QoC est quantifiée à l'aide de l'équation 5.8. La probabilité marginale du nœud QoS dans RB_2 et dans l'extension $RB_1 \oplus RB_2$ de la figure 5.20

est le résultat de l'inférence dans le module bayésien de la partie communication. Nous avons $[P(QoS = \text{bonne}); P(QoS = \text{dégradée}); P(QoS = \text{mauvaise})] = [0.81; 0.14; 0.05]$. La probabilité marginale du nœud QoC dans RB_2 est calculée et nous obtenons $[P(QoC = \text{bonne}); P(QoC = \text{dégradée}); P(QoC = \text{mauvaise})] = [0.24; 0.28; 0.48]$. Nous calculons chaque paramètre de la table de probabilités de la QoC du $RB_1 \oplus RB_2$ à l'aide de l'équation 5.8. Ils sont définis dans les tables 5.36, 5.37 et 5.38. Si nous prenons le cas où les états des nœuds sont tels que : $QoC=\text{bonne}$, $\text{état du robot}=\text{bon}$, $\text{contrôleur}=\text{bon}$ et $QoS=\text{dégradée}$:

$$P(QoC = b | R = b, C = b, QoS = \text{dé}) = \frac{P_1(QoC = b | R = b, C = b) \times \frac{P_2(QoC=b|QoS=dé)}{P_2(QoC=b)}}{\sum_{s=b,d,m} P_1(QoC = s | R = b, C = b) \times \frac{P_2(QoC=s|QoS=dé)}{P_2(QoC=s)}} \quad (5.11)$$

avec b : bon, dé : dégradé, m : mauvais, R : état du robot, C : contrôleur.

$$P(QoC = b | R = b, C = b, QoS = d) = \frac{0.93 \times \frac{0.23}{0.24}}{0.93 \times \frac{0.23}{0.24} + 0.064 \times \frac{0.31}{0.28} + 0.006 \times \frac{0.46}{0.48}} = 0.92$$

Finalement, le robot arrive à sa destination (30, 20) après 150s et reste stable, son orientation est stable et sa vitesse linéaire est nulle.

TABLE 5.36 – Table de probabilités conditionnelles de l'état de la QoC sachant un bon état du contrôleur

contrôleur	bon								
	b			dé			m		
état du robot	b	dé	m	b	dé	m	b	dé	m
QoS	b	dé	m	b	dé	m	b	dé	m
QoC=b	0.93	0.92	0	0.90	0.88	0	0	0	0
QoC=dé	0.06	0.07	0.43	0.09	0.11	0.42	0.95	0.96	0.58
QoC=m	0.01	0.01	0.57	0.01	0.01	0.58	0.05	0.04	0.42

5.7 Conclusion

Nous avons présenté dans ce chapitre une méthode de diagnostic distribué qui s'adapte à l'architecture du système. En effet, l'architecture du système évolue au cours de sa mission selon l'état de la QoS qui a une influence sur la QoC . Le diagnostic distribué consiste en un RB modulaire distribué entre la station de contrôle et le robot. Cette répartition des modules bayésiens dépend de la vision que chaque sous système a des conditions et

TABLE 5.37 – Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur dégradé

contrôleur	dégradé								
état du robot	b			dé			m		
QoS	b	dé	m	b	dé	m	b	dé	m
QoC=b	0.88	0.86	0	0.88	0.86	0	0.56	0.51	0
QoC=dé	0.11	0.12	0.28	0.11	0.12	0.28	0.33	0.37	0.17
QoC=m	0.01	0.02	0.72	0.01	0.02	0.72	0.11	0.12	0.83

TABLE 5.38 – Table de probabilités conditionnelles de l'état de la QoC sachant un état du contrôleur mauvais

contrôleur	mauvais								
état du robot	b			dé			m		
QoS	b	dé	m	b	dé	m	b	dé	m
QoC=b	0	0	0	0	0	0	0	0	0
QoC=dé	0	0	0	0	0	0	0	0	0
QoC=m	1	1	1	1	1	1	1	1	1

de l'environnement de fonctionnement. L'objectif de cet outil de diagnostic est de déterminer la cause de la dégradation de la QoC. En outre, le comportement stochastique des réseaux de communication justifie le choix de l'utilisation du RB et l'architecture modulaire et distribuée du RB semble pertinente pour un système distribué. Les premiers résultats de l'inférence et du scénario de diagnostic ont montré l'utilité de l'utilisation des RBs distribués modulaires. Ils ont aussi montré la capacité du RB développé à détecter la dégradation de la QoC et de la QoS et à superviser l'état du robot. Nous avons présenté un scénario où les deux modes de fonctionnements sont utilisés. Au début de la mission d'atteinte de cible, le RB modulaire a détecté la déviation du robot de sa trajectoire. Le robot commute ensuite sur un contrôleur embarqué jusqu'à ce qu'il trouve la nouvelle station de contrôle. L'aspect modulaire du RB a facilité la reconfiguration de l'outil de diagnostic selon l'architecture de commande ou de communication adaptée (RB distribué ou RB monolithique embarqué). C'est une approche qui montre bien l'importance du co-design dans ce travail. Cependant, la commutation entre les deux architectures de diagnostic a été difficile à mettre en œuvre. En effet, la procédure de vérification de l'assemblage des deux modules bayésiens de la partie commande et de la partie opérative, dans la procédure de commutation du diagnostic débarqué au diagnostic embarqué, n'est pas automatique. La vérification a été faite à l'avance. Nous avons simplifié, dans le modèle de simulation, cette procédure de commutation de telle sorte que la commutation est exécutée correctement. Aucun problème de pertes de données ou de retards ou de synchronisation n'est pris en compte.

L'intérêt des modules bayésiens développés est la possibilité d'utiliser les différents modules dans d'autres applications distribuées : les drones, systèmes industriels distribués, internet des objets etc. Dans ce cas, les modules de la partie commande et de la partie opérative sont mis à jour selon l'application utilisée. Dans le cas où un autre type de réseau de communication est utilisé, le module bayésien de la partie communication peut être réutilisé en modifiant les paramètres à superviser et les liens causaux. Les modules des parties commande et opérative restent alors intacts.

Conclusion et perspectives

Les SCRs sans fil sont déployés à l'heure actuelle dans de nombreuses applications. Ces applications induisent souvent de la mobilité, celle-ci aura des conséquences sur la maintenance et le diagnostic. Le comportement indéterministe des réseaux sans fil peut influencer la performance d'un système temps réel. Un tel type de système, s'il est implémenté via un réseau, a besoin de recevoir ses données dans un intervalle de temps bien défini avec entre autres un délai et une gigue maîtrisés. Les SCRs sans fil, considérés comme des systèmes temps réel, imposent l'adaptation de la QoS du réseau : bande passante ou débit, délai, gigue et taux de pertes de paquets. Par conséquent, la QoS doit pouvoir permettre de garantir un niveau acceptable de QoC.

Pour un SCR sans fil mobile, l'environnement perturbé et la mobilité limitent également la performance du système. Avec l'intégration du réseau dans la boucle de commande, la QoC est influencée par la QoS ; cette QoS donne une image de l'état d'un réseau qui peut avoir un comportement stochastique. Par conséquent, il n'y a pas de garantie d'avoir une bonne QoC. Il est parfois difficile de modéliser correctement l'impact de la QoS sur la QoC à cause de la dynamique des réseaux sans fil. La dégradation stochastique de la QoS peut ne pas être représentable par un modèle mathématique. Le recours aux méthodes probabilistes s'avère être une solution pour étudier le comportement des SCRs sans fil.

Les modèles probabilistes sont utilisés pour le diagnostic, le pronostic, la sûreté de fonctionnement, la maintenance, etc, et ils ont montré leur efficacité dans plusieurs applications. Les RBs en particulier, qui sont des modèles graphiques probabilistes, représentent les dépendances de cause à effet entre des événements dont on quantifie la fréquence d'apparition à l'aide des probabilités. Les RBs permettent d'effectuer un raisonnement de diagnostic. Il est alors possible d'identifier la cause de dégradation de la QoC : celle-ci peut être due soit à un problème lié à la QoS, soit à un problème lié à l'état des composants physiques du système, soit à un problème lié à l'état du contrôleur. Pour un robot mobile commandé en réseau par une station de contrôle en particulier, la QoC désigne la capacité du contrôleur à amener le robot à la position d'équilibre via le réseau tout en évitant les éventuels obstacles. Dans ce contexte, notre démarche a consisté à proposer un RB pour le diagnostic d'un robot mobile commandé en réseau dans le but de détecter la dégradation de la QoC et identifier la cause de la défaillance. Cette défaillance peut provenir soit du réseau, soit du robot, soit du contrôleur. Après avoir déterminé la cause de la dégradation de la QoC à l'aide du RB développé, une stratégie de prise de décision est établie pour garder un bon niveau de performance. Cette stratégie consiste en la commutation d'un contrôleur débarqué sur un contrôleur embarqué implémenté sur le robot.

Nous avons détaillé dans le chapitre 2 les problématiques d'un SCR sans fil et l'utilisation des réseaux sans fil dans des applications temps réel. Nous avons présenté un état de l'art sur les SCRs sans fil (WNCS) et nous avons détaillé l'influence des paramètres de

la QoS sur la QoC tels que le taux de pertes de paquet, le délai, la gigue, le débit et la connectivité. Nous avons ensuite présenté le modèle stochastique de référence de l'accès au canal de communication CSMA/CA, la chaîne de Markov de Bianchi [Bia00]. Nous avons positionné nos contributions par rapport à d'autres travaux.

Nous avons présenté dans le troisième chapitre les RBs, leurs caractéristiques et leurs applications dans différents types de raisonnement. Un état de l'art sur l'utilisation des RBs dans le diagnostic a été présenté. Nous avons analysé les types d'observations, en fonction du degré de certitude sur les informations reçues, qui peuvent être traités par l'algorithme d'inférence.

Nous avons considéré le robot mobile commandé via un réseau par une station distante comme étant un système distribué. En outre, l'architecture du système évolue en fonction du mode de fonctionnement actif : le robot est soit connecté à la station, soit autonome. Le système est conçu de sorte que la station possède les informations sur l'état du réseau et sur la mission du robot et par conséquent sur la QoC. Le robot, quant à lui, est connaisseur de son état interne, à savoir : de l'état de ses actionneurs, de l'état de ses capteurs et de l'état de sa batterie. Les RBs distribués modulaires sont alors pertinents pour répondre à la problématique du diagnostic distribué des systèmes commandés en réseau. Un état de l'art sur les différents RBs distribués a été fait en montrant à chaque fois les avantages et les inconvénients de chacun. L'architecture modulaire et distribuée des RBs de [LYC15] et [LLC16] a été choisie pour modéliser un système distribué dont l'architecture évolue au cours du temps et pour un robot dont les capacités de calcul et de mémoire sont limitées. La dépendance causale entre la QoS et la QoC implique que, dans le cas où le réseau sans fil n'assure pas le niveau de QoS requis, la QoC et la performance du système se dégradent. Nous avons défini, dans le quatrième chapitre, les limites des paramètres de la QoS au-delà desquelles la QoC n'est plus garantie. En analysant le modèle de Bianchi [Bia00] et ses limites, nous avons pu développer une autre extension adaptée à la modélisation du trafic temps réel des SCRs sans fil dans un réseau CSMA/CA. La période d'échantillonnage critique (la plus petite valeur que le réseau peut supporter), le débit de saturation correspondant à cette période, le délai d'aller retour moyen et la gigue maximale ont été déterminés à l'aide d'une chaîne de Markov. Cette chaîne de Markov modélise le trafic d'une boucle de commande : les paquets de commande et les paquets de mesures. Nous avons pu ensuite établir les expressions analytiques de certains paramètres de la QoS en fonction du nombre de boucles de commande qui partagent le même réseau. Nous avons étudié le cas du réseau IEEE 802.11b pour observer l'évolution des paramètres définis par les expressions analytiques, en fonction du nombre de boucles de commande et en faisant varier la taille de la fenêtre de contention initiale. Cela a permis de définir des seuils de performances qui sont utilisés dans le RB développé.

Dans le dernier chapitre, nous avons présenté la dernière contribution de la thèse qui consiste en une modélisation bayésienne du diagnostic d'un robot mobile commandé en réseau. C'est une structure bayésienne modulaire qui est composée de trois modules bayésiens

distribués entre la station et le robot. Le système global est divisé en trois parties : une partie communication, une partie opérative et une partie commande. Un module bayésien est associé à chaque partie pour surveiller son état. Pour l'apprentissage des probabilités du modèle bayésien, nous avons utilisé un banc de test, composé d'un robot Khepera 3 et d'une station. Après avoir défini les tables de probabilités conditionnelles, deux scénarios de diagnostic ont été présentés pour montrer la dynamique de l'inférence dans ce RB modulaire distribué pour détecter la dégradation de la QoS et la procédure de transmission de probabilités a posteriori via les liens virtuels entre les modules bayésiens. Quand la décision a été prise de commuter, du contrôleur débarqué sur un contrôleur embarqué, à cause d'une mauvaise QoS, le système change d'architecture et par conséquent de méthode de diagnostic. Nous avons établi une procédure de commutation d'un diagnostic distribué, reposant sur des RBs distribués modulaires, à un diagnostic embarqué, reposant sur un RB monolithique, et vice-versa. Nous avons ensuite appliqué cette procédure de commutation sur un modèle de simulation pour montrer que notre méthode permet au robot d'atteindre sa destination. La méthode de diagnostic a été appliquée aux deux cas de fonctionnements possibles (fonctionnement débarqué et fonctionnement embarqué). La commutation entre les contrôleurs a également été implémentée sur notre banc de tests.

Nous avons présenté dans ce mémoire une méthode de diagnostic pour un véhicule autonome qui permet de détecter une déviation du comportement prévu et de prendre des décisions pour maintenir une bonne QoS. Cette méthode est modulaire et évolue selon l'architecture du système. La commande embarquée peut subir des dégradations à cause des capacités limitées du robot en terme de mémoire, de puissance de calcul et pour optimiser la consommation d'énergie de la batterie du robot. Dès qu'une nouvelle station de contrôle est détectée, le robot doit s'associer à son réseau. Deux nouveaux modules bayésiens, un pour la commande et un pour la communication sont alors ajoutés sur la station de contrôle. La question qui se pose est de savoir si notre méthode est robuste et pertinente faces aux perturbations réseaux. Autrement dit, quels sont les impacts des pertes de données de diagnostic et des retards sur la pertinence des résultats de diagnostic obtenus par le RB? Une des perspectives de ce travail consiste à caractériser la robustesse de cette méthode de diagnostic face aux problèmes du réseau. Si une donnée de diagnostic sur l'état du nœud *état du robot* est perdue, le module bayésien de la partie opérative, qui est implémenté sur le robot, ne va pas la recevoir. La procédure de diagnostic dans la partie opérative ne va pas s'effectuer. Par conséquent, les informations sur les nouveaux états des composants du robot ne seront pas disponibles. Nous proposons aussi d'étudier, en termes de perspectives, le coût de commutation entre les contrôleurs. Dans le cas où le robot est face à plusieurs stations de contrôle, une fonction de coût va permettre de choisir la bonne station en établissant une stratégie de commutation optimale. Le RB modulaire développé va ensuite être adapté aux cas de plusieurs robots qui partagent le même réseau pour une stratégie de diagnostic collaboratif. Nous envisageons ensuite d'établir une version dynamique de ce RB modulaire (RB dynamique) pour le pronostic du système qui servira à

prévoir les dégradations de la QoS et de la QoC avant qu'elles n'apparaissent. Cela consiste à déterminer l'état des nœuds *QoC*, *QoS* et *état du robot* à un instant $t + 1$ sachant l'état de ces nœuds à l'instant t . Finalement, l'approche à base de modules bayésiens distribués peut trouver une utilisation pertinente dans le monde de l'internet des objets où différents objets mobiles, tantôt connectés, tantôt non connectés, peuvent :

- soit participer à un diagnostic plus global d'une infrastructure distribuée,
- soit récupérer via cette infrastructure des informations pour le diagnostic de l'objet proprement dit,
- soit effectuer localement le diagnostic en toute autonomie.

Dans ces différents cas de figure, les modules bayésiens pourront communiquer ou pas en fonction des contextes ou des besoins.

Inférence dans les réseaux Bayésiens

A.1 Raisonnement bayésien

Le raisonnement dans l'incertain manipule les croyances résultant d'informations ou d'évidences incomplètes et de connaissances faillibles. Il existe plusieurs sources d'incertitude que nous devons prendre en considération : l'ignorance (ou le manque de connaissances), le comportement aléatoire (ou indéterministe) et le manque de précision. La théorie de Bayes sert à comprendre une opinion ou une connaissance en prenant en compte les contraintes d'incertitude en utilisant des calculs de probabilités conditionnelles. La théorie de Bayes est une bonne représentation des connaissances et un bon outil de raisonnement dans l'intelligence artificielle. Le concept de probabilité conditionnelle est utilisé pour assister l'individu dans son raisonnement et sa prise de décision. Pour calculer la probabilité conditionnelle de X sachant Y , où X et Y sont deux variables non-indépendantes, telles que la variable X a une influence sur Y , on utilise le théorème de Bayes :

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)} = \frac{P(Y|X)P(X)}{P(Y)} \quad (\text{A.1})$$

Un autre concept de la théorie des probabilités qui va être utilisé dans le calcul d'inférence dans un RB est l'indépendance entre les variables. Une variable X indépendante d'une variable Y implique que :

$$P(X|Y) = P(X) \quad (\text{A.2})$$

La notion d'indépendance conditionnelle permet de simplifier le calcul dans des probabilités a posteriori et de définir le concept de d-séparation entre les variables d'un RB. Dans le cas d'indépendance conditionnelle, si les variables X et Y sont indépendantes sachant Z alors :

$$P(X|Y, Z) = P(X|Z) \quad (\text{A.3})$$

D'autres équations s'inspirent de la notion de la probabilité conditionnelle et sont fréquemment utilisées pour calculer les probabilités marginales. Ainsi, pour calculer la probabilité marginale d'une variable X qui dépend de plusieurs variables Y_i , l'équation A.4 est utilisée :

$$P(X) = \sum_i P(X|Y_i) \quad (\text{A.4})$$

Pour appliquer l'équation A.1 de Bayes dans un raisonnement sur un RB, redéfinissons les variables X et Y autrement. Soit une hypothèse h sur une variable X ($X = h$) conditionnée par une évidence e sur Y ($Y = e$), l'équation A.1 devient :

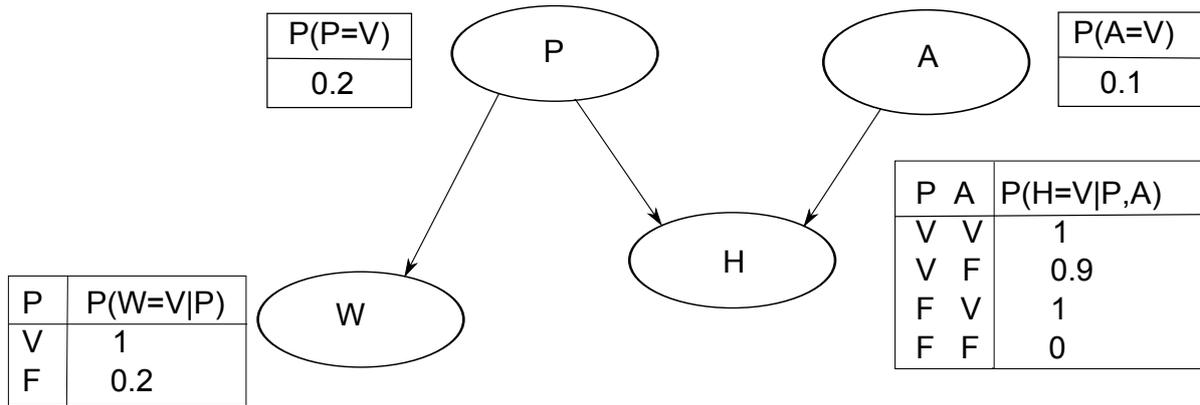
$$P(h|e) = \frac{P(e|h)P(h)}{P(e)} \quad (\text{A.5})$$

L'équation A.5 décrit un rapport entre la vraisemblance $P(e|h)$ de l'évidence e multipliée par la probabilité a priori $P(h)$ de h et un terme de normalisation $P(e)$ (ce terme de normalisation garantit que la somme de toutes les probabilités de h est égale à 1). La probabilité $P(h|e)$ n'est que la probabilité a posteriori après avoir appliqué une évidence e , c'est la nouvelle croyance sur la variable hypothèse h .

Un RB est un graphe probabiliste qui représente des relations de dépendance de cause à effet entre des variables comme cela a été détaillé dans le chapitre 3. C'est un graphe acyclique où des nœuds sont liés par des arcs orientés comme le montre la figure A.1 [Pea88]. Un nœud discret peut avoir des valeurs booléennes $\{\text{vrai, faux}\}$, des valeurs ordonnées, par exemple $\{\text{bon, dégradé, mauvais}\}$, ou des valeurs numériques. Les relations entre les nœuds sont quantifiées par des probabilités. Les nœuds racines sans parents possèdent des probabilités marginales a priori. Quant aux nœuds fils, on leur associe une table de probabilités conditionnelles sachant l'état de leurs parents comme le montre la figure A.1. Un nœud X est parent d'un nœud fils Y s'il existe un arc orienté allant de X à Y . Dans notre exemple de la figure A.1, relatif à de M. Holmes et son jardin, le nœud P est le parent du nœud H . Une autre propriété qui est utile et qui sert aussi à optimiser le calcul dans l'algorithme d'inférence est la limite ou la couverture de Markov (*Markov Boundary*). Pour une variable V , la limite de Markov est constituée de ses nœuds parents, ses nœuds fils et leurs nœuds parents. Une variable V est conditionnellement indépendante des autres variables qui sont en dehors de l'ensemble de variables de sa couverture de Markov. Cela veut dire que si les nœuds de cet ensemble (limite de Markov de V) sont observés, ils bloquent le passage d'informations entre les autres nœuds extérieurs à cette limite et le nœud de la variable V .

A.2 Inférence exacte dans une structure en arbre

La procédure d'inférence dans un RB sert à calculer les probabilités a posteriori des variables non observées sachant un ensemble d'évidences. Cela consiste à mettre à jour la valeur de la croyance sur une variable en appliquant des évidences sur n'importe quel autre nœud. L'algorithme d'inférence est basé sur le théorème de Bayes sachant qu'il existe une inférence exacte et une inférence approximative qui optimise le temps d'inférence dans les RBs de grandes tailles et plus compliqués. Nous allons commencer par un exemple simple pour montrer la procédure d'inférence et la mise à jour des valeurs de croyances. Prenons



A : M. Holmes a oublié de débrancher l'arroseur automatique
 P : il a plu pendant la nuit
 H : l'herbe du jardin de M. Holmes est humide
 W : l'herbe du jardin de M. Watson est humide

FIGURE A.1 – Exemple d'un RB [Pea88]

le cas d'un RB composé de deux nœuds X et Y dont le sens de la relation de dépendance est $X \rightarrow Y$. Si le nœud Y subit une évidence telle que $Y = y$, la valeur de probabilité a posteriori de X est calculée en utilisant l'équation A.5 :

$$P(X = x|Y = y) = \frac{P(Y = y|X = x)P(X = x)}{P(Y = y)} \tag{A.6}$$

$$= \alpha P(X = x)\lambda(x) \tag{A.7}$$

où $P(X = x)$ est la probabilité a priori, $\lambda(x) = P(Y = y|X = x)$ est la vraisemblance et α est une constante de normalisation (une constante déterminée sachant que la somme des croyances sur X doit être égale à 1).

Pour une structure bayésienne plus compliquée, [Pea88] a établi un algorithme de propagation de messages que nous allons voir dans la sous-section suivante.

A.2.1 Algorithme de propagation de messages de Pearl

Le premier algorithme d'inférence exacte a été établi par Pearl, c'est un algorithme limité aux structures en arbre ou forêt d'arbres [Pea88]. Dans la procédure d'inférence [Pea88], un nœud X est un producteur de deux types de messages : des messages λ qu'il envoie à ses parents et les messages π qu'il transmet vers ses nœuds fils comme le montre la figure A.2. Tous les nœuds du RB échangent des messages avec leurs voisins jusqu'à ce

qu'un équilibre soit atteint. Soit le RB de la figure A.2 où le nœud X possède m parents

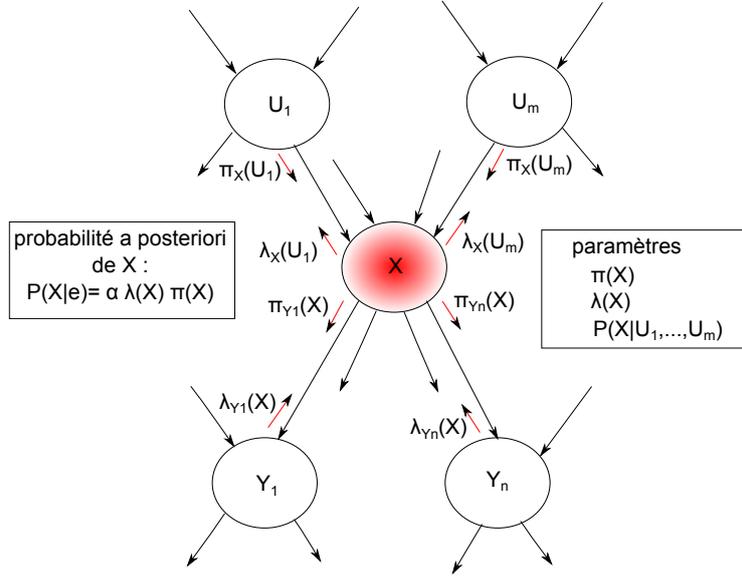


FIGURE A.2 – Propagation de messages entre les nœuds

U_1, \dots, U_m et n enfants Y_1, \dots, Y_n et une évidence e . L'algorithme consiste à mettre à jour la probabilité a posteriori $P(X|e)$ de X en utilisant les paramètres $\lambda(X)$ et $\pi(X)$ et l'équation suivante :

$$P(X|e) = \alpha \lambda(X) \pi(X) \quad (\text{A.8})$$

Les deux paramètres $\lambda(X)$ et $\pi(X)$ sont calculés en utilisant les messages λ et π provenant des parents et des enfants de X . Un message π est envoyé dans le sens de l'arc orienté, du parent vers l'enfant, tel que $\pi_{\text{récepteur}}(\text{expéditeur}) = \pi_{\text{enfant}}(\text{parent})$. Un message λ est envoyé dans le sens contraire des arcs orientés, de l'enfant au parent, tel que $\lambda_{\text{expéditeur}}(\text{récepteur}) = \lambda_{\text{enfant}}(\text{parent})$. Soit $\pi_X(U_i) = P(U_i|E_{U_i \setminus X})$ avec $E_{U_i \setminus X}$ l'ensemble de toutes les évidences connectées à U_i sauf via X . Soit $\lambda_{Y_j}(X) = P(E_{Y_j \setminus X}|X)$ avec $E_{Y_j \setminus X}$ l'ensemble des évidences connectées à Y_j sauf via X . Pour calculer les probabilités a posteriori $P(X = x_i|e)$, le nœud X collecte les paramètres $\pi_X(U_i)$ et $\lambda_{Y_j}(X)$, les équations suivantes sont alors utilisées :

$$\lambda(x_i) = \begin{cases} 1 & \text{si l'évidence est } X = x_i \\ 0 & \text{si l'évidence est } X = x_j \text{ et } x_j \neq x_i \\ \prod_j \lambda_{Y_j}(x_i) & \text{sinon} \end{cases} \quad (\text{A.9})$$

et

$$\pi(x_i) = \sum_{u_1, \dots, u_m} P(x_i|u_1, \dots, u_m) \prod_i \pi_X(u_i) \quad (\text{A.10})$$

Le nœud X envoie ensuite les nouveaux messages λ à ses parents tel que :

$$\lambda_X(u_i) = \sum_{x_i} \lambda(x_i) \sum_{u_k, k \neq i} P(x_i | u_1, \dots, u_m) \prod_{k \neq i} \pi_X(u_k) \quad (\text{A.11})$$

Le nœud X envoie également les nouveaux messages π à ses enfants :

$$\pi_{Y_j}(x_i) = \begin{cases} 1 & \text{si l'évidence est } X = x_i \\ 0 & \text{si l'évidence est } X = x_j \text{ et } x_j \neq x_i \\ \alpha \pi(x_i) \prod_{k \neq j} \lambda_{Y_k}(x_i) & \text{sinon} \end{cases} \quad (\text{A.12})$$

A.2.1.1 Exemple de l'application de l'algorithme de Pearl

Nous reprenons le RB de la figure A.1, avant d'appliquer des observations, nous calculons les probabilités marginales a priori des nœuds W et H à l'aide de l'algorithme de propagation de message de Pearl. Le nœud P envoie à son nœud enfant W un message π que nous calculons en utilisant l'équation A.12 :

$$\pi_W(P) = (0.2; 0.8) \quad (\text{A.13})$$

Nous calculons ensuite $\pi(W)$ en utilisant la formule de l'équation A.10 :

$$\pi(W) = (0.2 * 1 + 0.8 * 0.2; 0.2 * 0 + 0.8 * 0.8) = (0.36; 0.64) \quad (\text{A.14})$$

Nous calculons également de la même façon les messages π envoyés par les nœuds parents P et A à leur nœud enfant H :

$$\pi_H(P) = (0.2; 0.8) \quad (\text{A.15})$$

$$\pi_H(A) = (0.1; 0.9) \quad (\text{A.16})$$

Nous obtenons par la suite $\pi(H)$ en utilisant l'équation A.10 :

$$\begin{aligned} \pi(H) &= (0.2 * 0.1 * 1 + 0.8 * 0.1 * 0.9 + 0.2 * 0.9 * 1 + 0.8 * 0.9 * 0; \\ &\quad 0.2 * 0.1 * 0 + 0.8 * 0.1 * 0.1 + 0.2 * 0.9 * 0 + 0.8 * 0.9 * 1) \\ &= (0.272; 0.728) \end{aligned} \quad (\text{A.17})$$

Les probabilités marginales des nœuds W et H sont ensuite calculées à l'aide de l'équation suivante :

$$P(X) = \alpha \lambda(X) \pi(X) \quad (\text{A.18})$$

Nous obtenons :

$$P(W) = [0.36; 0.64] \quad (\text{A.19})$$

$$P(H) = [0.272; 0.728] \quad (\text{A.20})$$

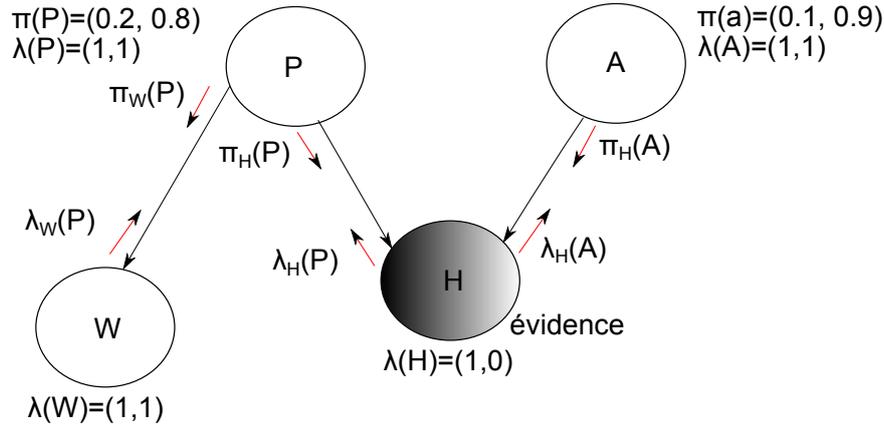


FIGURE A.3 – Exemple de propagation de messages avec une phase d'initiation

Le nœud H subit une évidence certaine ($P(H = vrai) = 1$). Cela veut dire que M.Holmes a remarqué que l'herbe de son jardin est humide. Cette observation/évidence certaine déclenche la propagation des messages comme le montre la figure A.3. L'évidence est caractérisée par $\lambda(H) = (1, 0)$ qui doit être propagée dans le réseau. Les valeurs des paramètres $\lambda_H(P)$ et $\lambda_H(A)$ sont calculées en utilisant l'équation A.11 et envoyées aux nœuds P et A .

$$\begin{aligned}
 \lambda_H(P) &= \left(\sum_{H=V,F} \lambda(H) \sum_A P(H|P=V, A) \pi_H(A); \right. \\
 &\quad \left. \sum_{H=V,F} \lambda(H) \sum_A P(H|P=F, A) \pi_H(A) \right) \\
 &= (1 \times (1 \times 0.1 + 1 \times 0.9) + 0 \times (0 \times 0.1 + 0 \times 0.9); \\
 &\quad 1 \times (0.9 \times 0.1 + 0 \times 0.9) + 0 \times (0.1 \times 0.1 + 1 \times 0.9)) \\
 &= (1; 0.09)
 \end{aligned} \tag{A.21}$$

Nous calculons de la même façon $\lambda_H(A)$:

$$\begin{aligned}
 \lambda_H(A) &= \left(\sum_{H=V,F} \lambda(H) \sum_P P(H|A=V, P) \pi_H(P); \right. \\
 &\quad \left. \sum_{H=V,F} \lambda(H) \sum_P P(H|A=F, P) \pi_H(P) \right) \\
 &= (1 \times (1 \times 0.2 + 0.9 \times 0.8) + 0 \times (0 \times 0.2 + 0.1 \times 0.8); \\
 &\quad 1 \times (1 \times 0.2 + 0 \times 0.8) + 0 \times (0 \times 0.2 + 1 \times 0.8)) \\
 &= (0.92; 0.2)
 \end{aligned} \tag{A.22}$$

Après avoir reçu un message λ de son nœud fils H , le nœud P envoie un message π à son nœud fils W . En utilisant l'équation A.12, nous obtenons :

$$\pi_W(P) = \alpha \pi(P) \lambda_H(P) \tag{A.23}$$

et finalement $\pi_W(P) = (0.7353; 0.2647)$. Pour calculer $\pi(W)$, nous utilisons l'équation A.10 :

$$\begin{aligned}\pi(W) &= \left(\sum_{P=V,F} P(W=V|P)\pi_W(P); \sum_{P=V,F} P(W=F|P)\pi_W(P) \right) \\ &= (1 \times 0.7353 + 0.2 \times 0.2647; 0 \times 0.7353 + 0.8 \times 0.2647) \\ &= (0.7882; 0.2118)\end{aligned}\tag{A.24}$$

Les probabilités a posteriori de chaque nœud du RB sont calculées en utilisant l'équation A.8. Nous résumons toutes les valeurs obtenues des paramètres et des probabilités dans le tableau A.1.

TABLE A.1 – Tableau récapitulatif

	probabilité a posteriori $P(X e)$	$\pi(x)$	$\lambda(x)$
P	(0.7353; 0.2647)	(0.2; 0.8)	(1; 0.09)
A	(0.3382; 0.6618)	(0.1; 0.9)	(0.92; 0.2)
W	(0.7882; 0.2118)	(0.7882; 0.2118)	(1; 1)
H	(1; 0)	(0.272; 0.728)	(1; 0)

A.3 Inférence avec évidence virtuelle

Nous avons détaillé dans la sous section 3.4.2.1 du chapitre 3 le calcul de la probabilité a posteriori dans le cas d'une observation incertaine ou virtuelle. L'inférence d'évidences virtuelles nécessite l'utilisation d'un nœud virtuel. Ce nœud est le fils du nœud qui subit une évidence virtuelle comme le montre la figure A.4 où V représente l'évidence virtuelle sur E . Le nœud virtuel V n'appartient pas à la structure initiale du RB et par conséquent dans la procédure de propagation de messages de Pearl, il ne possède pas de paramètre $\lambda(V)$ mais il envoie un message $\lambda_V(A)$ à son nœud père A comme le montre la figure A.4.

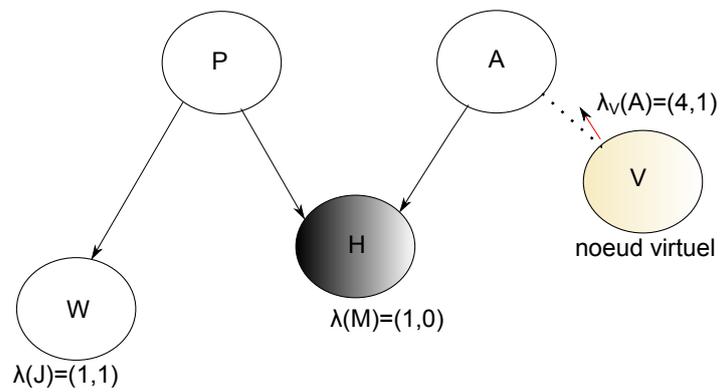


FIGURE A.4 – Exemple de propagation de messages avec une évidence virtuelle

Plateforme de tests et de simulation

B.1 Présentation du robot Khepera 3

Le robot Khepera 3 est un véhicule autonome qui fait partie de la famille des robots mobiles miniatures. Conçu à l'EPFL (Ecole Polytechnique Fédérale de Lausanne), le robot Khepera 3 (voir figure B.1) possède une puissance de calcul et une capacité de mémoire limitées, de multiples réseaux de capteurs pour la détection à la fois longue distance et à courte portée de l'objet, un système de batterie pour une autonomie optimale, et un bon lecteur d'odométrie.



FIGURE B.1 – Robot Khepera 3

B.1.1 Moteurs

Le robot est actionné par deux moteurs à courant continu indépendants. Les deux moteurs sont commandés grâce à une modulation en largeur d'impulsions (PWM) générée par le micro-contrôleur PIC18F4431. D'après les études faites par [Mec10], ces moteurs sont contrôlés par un régulateur PI. Afin de choisir les bons coefficients pour le correcteur (K_p pour le terme proportionnel et T_i pour l'action intégrale), [Mec10] a mis le moteur en boucle ouverte et il l'a soumis à un échelon de tension. D'après la réponse du moteur, le système est un système de premier ordre d'équation B.1. La période échantillonnage

utilisée est $T'_e = 20ms$ et est définie par rapport au temps de réponse de la boucle de régulation de la vitesse angulaire du moteur. Ce temps de réponse est égal à $100ms$ et $T'_e = \frac{\text{temps de réponse}}{5}$ en respectant l'inégalité 2.2.

$$H = \frac{K}{\tau \times s + 1} \quad (\text{B.1})$$

où H est la fonction de transfert, K ($= 6.13$) le gain du moteur et τ ($= 0.71s$) la constante de temps.

B.1.2 Encodeurs

Chaque moteur possède son propre codeur incrémental intégré qui est situé sur l'axe du moteur. Ces encodeurs fournissent environ 54 impulsions tous les 10 millimètres parcourus. Étant donné que le diamètre de la roue est de 41mm, chaque tour de roue fait avancer le robot de 128,8mm. Les encodeurs du Khepera 3 étant réglés sur une résolution x4, nous avons donc, pour chaque roue, un encodeur d'une résolution de 2764 points par tour ce qui permet d'avoir une précision d'environ 50 micromètres. Nous utilisons ces encodeurs afin de connaître en permanence la position du robot par rapport à une origine. Le fonctionnement de ces dispositifs est montré dans la figure B.2. Pour pouvoir connaître, à tout

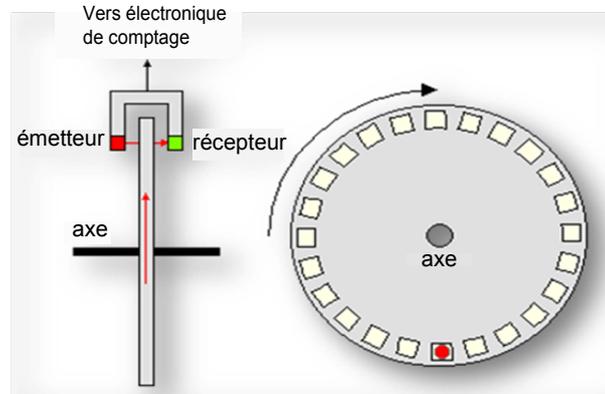


FIGURE B.2 – Encodeur

moment, la position du robot, nous utilisons le principe de l'odométrie. Cette technique consiste à calculer la position et l'orientation du robot à chaque instant en cumulant tous les déplacements effectués depuis la dernière position connue par les deux roues. Une démonstration de ce principe est explicitée dans la figure B.3. Soit $(X, Y, \theta) = (X_0, Y_0, \theta_0)$ la position initiale du robot, la variation de l'angle, $d\theta$, est définie par l'équation B.2 :

$$d\theta = \tan(d\theta) = \frac{d\epsilon_r - d\epsilon_l}{l} \implies \theta = \theta_0 + d\theta \quad (\text{B.2})$$

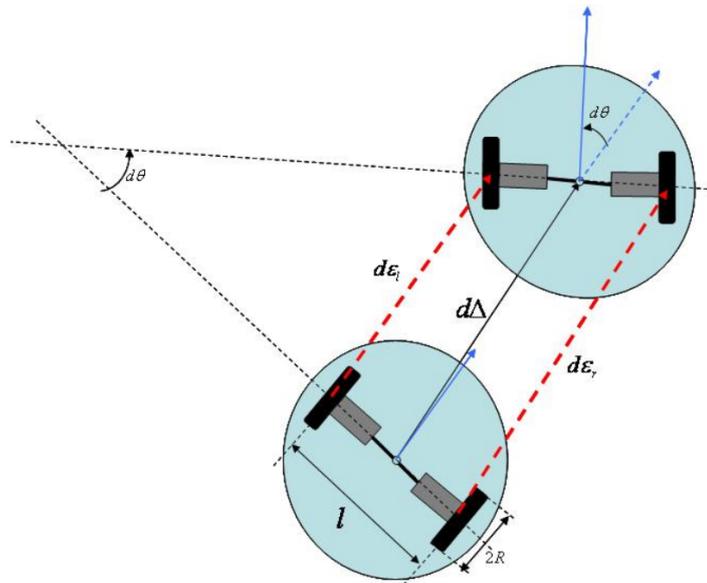


FIGURE B.3 – Principe d'odométrie

Quant à la variation de déplacement du robot, elle est déterminée par l'équation B.3 :

$$d\Delta = \frac{d\epsilon_r + d\epsilon_l}{2} \quad (\text{B.3})$$

Les variations de positions sont définies par les équations suivantes :

$$dx = d\Delta \cos(d\theta) \quad (\text{B.4})$$

$$dy = d\Delta \sin(d\theta) \quad (\text{B.5})$$

La nouvelle position du robot est calculée avec les équations suivantes :

$$x = x_0 + dx \quad (\text{B.6})$$

$$y = y_0 + dy \quad (\text{B.7})$$

B.1.3 Capteurs de détection d'obstacles

Le robot possède 11 capteurs infrarouges *TCRT5000* positionnés comme le montre la figure B.4. Neuf de ces capteurs sont répartis autour du robot pour permettre la détection et l'évitement d'obstacle, et deux autres en-dessous du robot, pouvant être utilisés pour suivre une ligne ou tout simplement détecter la présence du sol. Ces capteurs sont composés d'un émetteur et d'un récepteur (photo-transistor). Ces capteurs peuvent effectuer deux

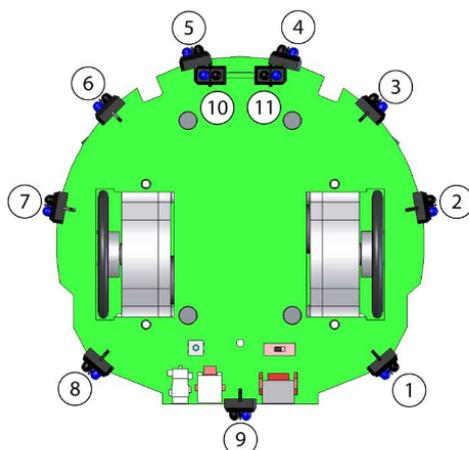


FIGURE B.4 – Capteurs infrarouges

types de mesures : la détection de la lumière ambiante et la détection des obstacles. Pour le banc test développé par [Leg17], ils ne sont utilisés que pour le dernier cas. L'efficacité de cette détection d'obstacles dépend de la nature et de la couleur des obstacles. En effet, ces capteurs sont capables de détecter la présence d'un objet blanc jusqu'à 12cm . Cependant, plus l'objet est sombre et plus la distance de détection est réduite. Chaque capteur renvoie des signaux convertis en un nombre compris entre 0 et 4000 suivant la distance et la couleur de l'objet. Le tableau B.1 montre les données fournies par ces capteurs en fonction de la distance et de la couleur de l'objet détecté. Ils ont néanmoins plusieurs inconvénients. Le premier est leur sensibilité aux fortes sources de lumière qui contiennent un fort rayonnement infrarouge, aux couleurs et à la nature des obstacles. Le deuxième inconvénient est la non-linéarité de leur réponse en fonction du flux (signaux). Leur temps de réponse est en général élevé et leur bande passante est limitée. La mesure est effectuée toutes les 33ms . Pendant les 33ms , les onze capteurs sont lus de manière séquentielle toutes les 3ms en commençant par le capteur arrière gauche (capteur 8 sur la figure B.4) et en poursuivant dans le sens horaire. Chaque mesure se traduit par une valeur analogique convertie en une valeur codée sur 12 bits pour être interprétée par le contrôleur.

B.1.4 Protocoles de communication

La communication interne au robot se fait par liaison série I2C. L'envoi de la commande, du programme et la communication entre l'ordinateur et le robot peut se faire soit par liaison série, par Bluetooth ou par Wifi 802.11b/g . Pour le banc de test développé par [Leg17], c'est la communication Wifi qui est mise en œuvre.

TABLE B.1 – Valeurs renvoyées par les capteurs en fonction de la distance et de la couleur de l'obstacle

valeurs	distance (cm)		distance moyenne (cm)
	obstacle blanc	obstacle rouge	
0	-	-	-
500	10	4.1	7.05
1000	6.7	3	4.5
1500	5	2.6	3.8
2000	3	2.2	2.6
2500	2	2	2
3000	1.4	1.5	1.45
3500	0.9	0.9	0.9
4000	0	0.5	0.25

B.1.5 Modèle cinématique

Dans le but d'implémenter l'architecture de commande, il est indispensable de décrire le modèle cinématique pour que le robot arrive à accomplir sa mission. Le robot utilisé dans l'étude effectuée, schématisé dans la figure B.5, est unicycle. Cela signifie qu'il est actionné par deux roues indépendantes. C'est un robot non holonome car il est impossible de le faire déplacer dans une direction perpendiculaire aux roues motrices. Pour l'étude cinématique

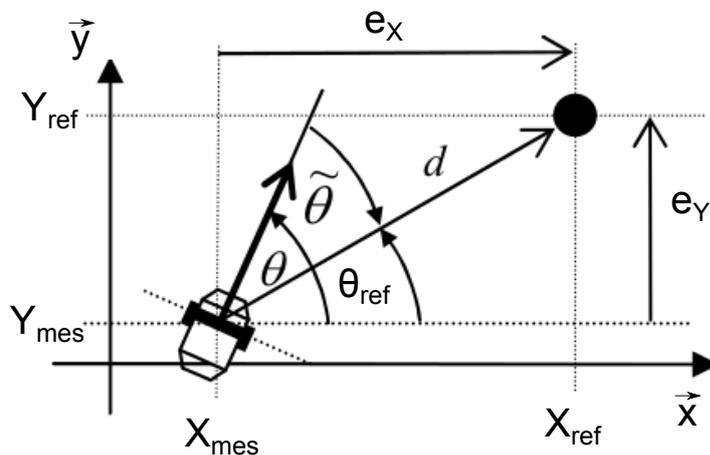


FIGURE B.5 – Modèle cinématique du robot

du robot, définie dans [TCK07], la contrainte non holonome implique un roulement sans glissement d'un point de l'axe de la roue. Les variables d'état du robot sont x , y et θ , qui désignent les coordonnées du centre de masse du robot dans le repère $R(O, \vec{x}, \vec{y})$, et

l'angle θ , qui représente l'orientation de l'axe de robot par rapport à l'axe (O, \vec{x}) . Le modèle cinématique est donné par le système non linéaire B.8 suivant qui représente les équations d'état du système :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (\text{B.8})$$

où les vitesses linéaire et angulaire, notées v et ω respectivement, sont les lois de commandes (les entrées).

B.1.6 Modèle dynamique

Le modèle dynamique du robot est obtenu par les équations de transfert des deux moteurs. Chaque fonction de transfert a été établie par identification [Mec10] :

$$\frac{\omega_{\text{droite/gauche}}}{U} = \frac{K}{1 + \tau \times s} \quad (\text{B.9})$$

où U représente la tension appliquée à chaque moteur et $\omega_{\text{droite/gauche}}$ la vitesse angulaire de chaque moteur.

B.2 Architecture de commande du robot

Dans le but d'atteindre une cible, définie par ses coordonnées, deux contrôleurs sont conçus et implémentés. Le premier contrôleur concerne la commande des deux roues en vitesse angulaire en utilisant un contrôleur PI pour chacune des deux roues motrices. Le deuxième contrôleur est un contrôleur de position qui calcule les commandes en vitesse linéaire v et angulaire ω du robot en utilisant les lois de commande non linéaire proposées par [TCK07]. Deux types d'architectures de commande sont conçus pour le système : un contrôleur embarqué et un contrôleur débarqué. En effet, le robot peut être autonome et dans ce cas tout calcul nécessaire est effectué au niveau de son micro-contrôleur embarqué, ce qui veut dire que le contrôleur de position est implémenté sur le robot. En mode débarqué, le robot est connecté à une station distante et le contrôleur de position est implémenté sur la station de contrôle.

B.2.1 Commande en vitesses

Dans le but d'avoir une bonne commande de vitesses, un régulateur PI est utilisé au niveau de chaque roue. Les entrées v et ω représentent les vitesses de référence, calculées

par le contrôleur implémenté sur la station (voir figure 5.1) pour le robot. Ces vitesses doivent être transformées en vitesses applicables à chaque roue du robot, c'est-à-dire en vitesses angulaires droite et gauche ω_{droite} et ω_{gauche} . Ces vitesses de référence, qui sont les consignes pour le contrôleur local, sont définies par les équations B.10 et B.11 et calculées à partir de v et ω .

$$\omega_{droite} = \frac{v + \frac{\omega \times l}{2}}{R} \quad (\text{B.10})$$

$$\omega_{gauche} = \frac{v - \frac{\omega \times l}{2}}{R} \quad (\text{B.11})$$

où l est la distance qui sépare les deux roues et R est le rayon des roues.

Chaque contrôleur local (voir figure 5.2) implémenté sur le robot est un régulateur PI dont la fonction de transfert est définie par l'équation B.12 :

$$C(s) = K_p + \frac{1}{T_i \times s} \quad (\text{B.12})$$

Pour identifier les coefficients du régulateur, les moteurs sont soumis à un échelon de vitesse. La procédure d'identification a permis d'obtenir les gains $K_p = 3.5$ et $T_i = 0.2$ pour un temps de réponse $T_r = 100ms$. En passant au régulateur PI discret, la période d'échantillonnage de la boucle locale choisie est $T_e' = 20ms$ pour respecter la condition de l'inégalité 2.2. Ainsi, pour une période d'échantillonnage $T_e' = 20ms$, les gains du PI discret sont $K_{pk} = 3.5$ et $T_{ik} = 0.1$.

La réponse du robot aux consignes (vitesses linéaire v et angulaire ω) est obtenue à partir des vitesses angulaires mesurées au niveau des moteurs du robot. Les réponses en vitesse linéaire v^* et en vitesse angulaire ω^* sont définies par les équations suivantes :

$$v^* = R \times \frac{\omega_{droite}^* + \omega_{gauche}^*}{2} \quad (\text{B.13})$$

$$\omega^* = R \times \frac{\omega_{droite}^* - \omega_{gauche}^*}{l} \quad (\text{B.14})$$

où ω_{droite}^* et ω_{gauche}^* sont les vitesses angulaires mesurées.

B.2.2 Commande de position

Le robot doit atteindre la position finale $(X_{ref}, Y_{ref}, \theta_{ref})$ en partant d'une position initiale (X_i, Y_i, θ_i) . Pour cela, il faut calculer les erreurs d'orientation $\tilde{\theta}$ et de position d . L'erreur de position d est calculée selon l'équation B.15 :

$$d = \sqrt{e_x^2 + e_y^2} \quad (\text{B.15})$$

avec $e_X = X_{ref} - X_{mes}$, $e_Y = Y_{ref} - Y_{mes}$ où X_{mes} , Y_{mes} sont les positions mesurées du robot.

L'erreur d'orientation $\tilde{\theta}$ est définie par l'équation B.16 :

$$\tilde{\theta} = \theta_{ref} - \theta_{mes} = \tan^{-1}\left(\frac{e_X}{e_Y}\right) - \theta_{mes} \quad (\text{B.16})$$

Le contrôleur non linéaire de position, fournit les lois de commandes en vitesses qui sont définies dans les équations B.17 et B.18 :

$$v = \frac{v_{max}}{1 + |d|} d \cos(\tilde{\theta}) \quad (\text{B.17})$$

$$v = \frac{v_{max}}{1 + |d|} d \cos(\tilde{\theta}) \sin(\tilde{\theta}) + K_{\tilde{\theta}} \tanh(K_{\theta} \tilde{\theta}) \quad (\text{B.18})$$

où d est l'erreur de position définie dans l'équation B.15, v_{max} est la vitesse maximale du robot telle que $v_{max} = 0,33m/s$, et $K_{\tilde{\theta}} = 0,5$ et $K_{\theta} = 1$ sont des constantes définies dans [TCK07].

[TCK07] a démontré que le système est stable au point d'équilibre $\begin{pmatrix} d \\ \tilde{\theta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ en utilisant la fonction de Lyapunov candidate définie dans l'équation B.19 :

$$V = \frac{\tilde{\theta}^2}{2} + \frac{d^2}{2} \quad (\text{B.19})$$

La fonction de Lyapunov V de l'équation B.19 est définie positive. La dérivée de la fonction candidate est :

$$\dot{V} = \tilde{\theta} \dot{\tilde{\theta}} = \tilde{\theta} \left(\frac{v \sin(\tilde{\theta})}{d} - \omega \right) - dv \cos(\tilde{\theta}) \quad (\text{B.20})$$

En remplaçant v et ω par leurs formules B.17 et B.18, la dérivée de la fonction est bien définie négative.

B.3 Scénarios d'utilisation du banc de test

Les travaux de [Leg17] consistent à implémenter l'architecture de commande simulée et à tester les performances du robot commandé. Des perturbations influencent la mission du robot : des problèmes liés au réseau ou des problèmes internes au robot tels que la dégradation de l'état des moteurs et de ses capteurs détectant les obstacles. Le robot et la station de contrôle communiquent via un réseau sans fil IEEE 802.11b avec un débit de communication de $11Mbit/s$. Le contrôleur envoie le paquet de commande de 130 octets au robot et le robot envoie en retour le paquet de mesures.

La mobilité du robot a un impact sur la performance du système. En effet, quand le

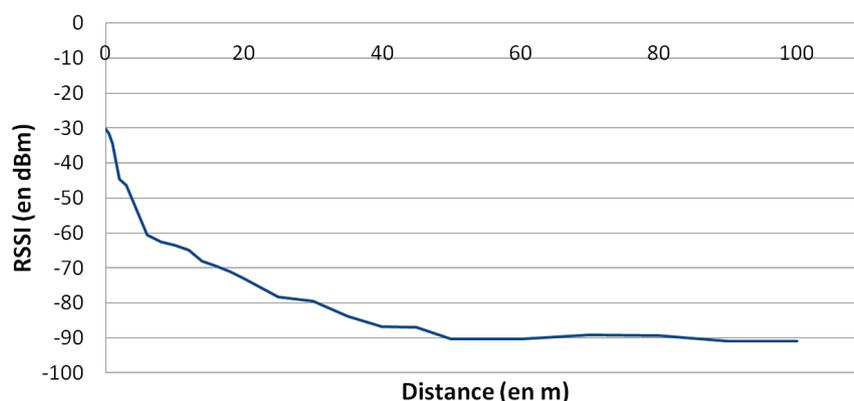


FIGURE B.6 – Évolution du RSSI en fonction de la distance

robot s'éloigne de la station de contrôle, la qualité du signal reçu se dégrade impactant la bonne réception des paquets échangés. La figure B.6 représente l'évolution du *RSSI* en fonction de la distance. Le *RSSI* diminue quand la distance qui sépare le robot et la station augmente. Les expérimentations menées dans [Leg17] ont montré l'impact de la qualité de la connectivité entre la station et le robot, mesurée par le *RSSI*, sur la QoC que nous résumons dans le tableau B.2. Par ailleurs, l'indéterminisme du réseau dégrade

TABLE B.2 – Impact de la qualité du signal reçu sur la QoC

QoC	RSSI(dBm)	distance(m)	taux de pertes de paquets
bonne	$RSSI \geq -80$	$distance \leq 30$	2.38%
dégradée	$-85 < RSSI < -80$	$30 < distance < 50$	40.21%
mauvaise	$RSSI \leq -85$	$distance \geq 50$	78.72%

la mission du robot. Le contrôleur d'évitement d'obstacles (développé dans [TCK07] et implémenté par [Leg17] sur Khepera), est moins performant vis-à-vis des perturbations du réseau. La figure B.7 montre l'impact des pertes de paquets sur la mission d'évitement d'obstacle. Le robot part de la position initiale $(X_0, Y_0, \theta_0) = (0, 0, 0)$ pour arriver à une position finale $(1, 1)$ en évitant un obstacle. La figure B.7 montre le comportement du robot pour différents taux de pertes de paquets. Quand il n'y a pas de pertes de paquets, le robot évite bien l'obstacle. À 50% de pertes, le robot percute l'obstacle vu qu'il ne reçoit pas toutes les données de vitesses qui lui servent à éviter l'obstacle. À partir d'un taux moyennement élevé de pertes, par exemple pour des pertes de 66%, le robot heurte l'obstacle deux fois. À partir de 75% de pertes, le robot rencontre une grande difficulté pour éviter l'obstacle comme le montre la figure B.7. D'autres expérimentations ont été faites sur les moteurs. La dégradation de l'état des moteurs influence la performance de la mission du robot malgré les corrections du régulateur PI. Pour chaque moteur, trois états sont définis suivant le pourcentage de l'erreur de consigne par rapport à la consigne en vitesse angulaire

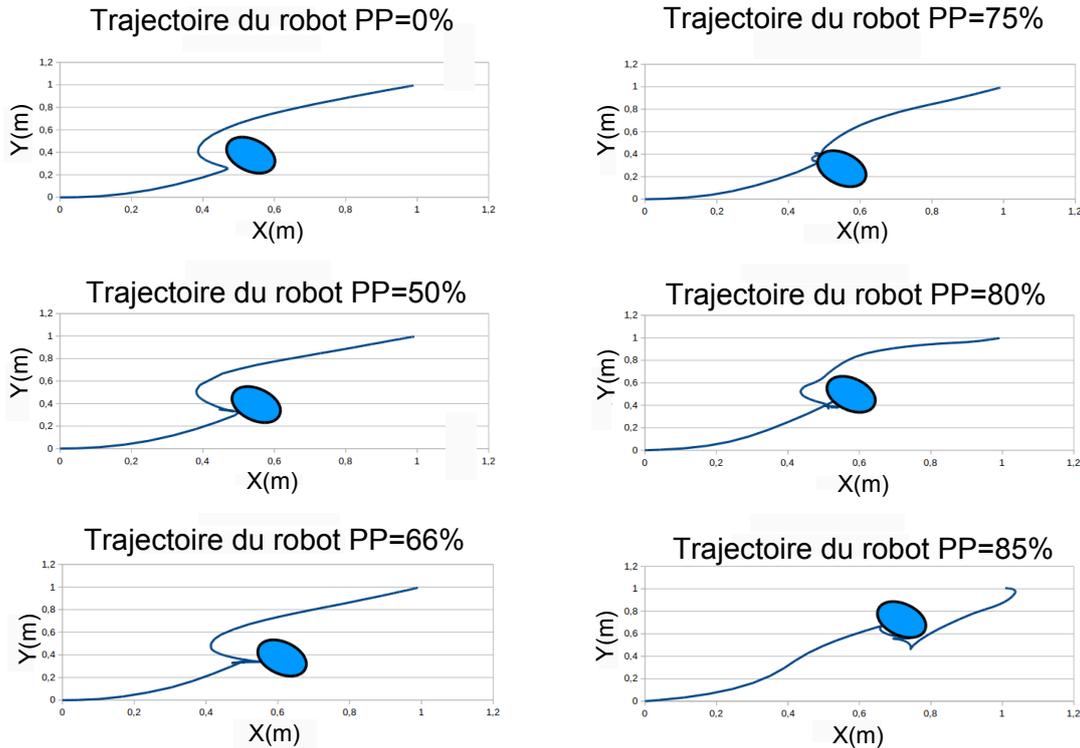


FIGURE B.7 – Évitement d'obstacle en fonction du taux de pertes de paquets

(voir tableau 5.2). Pour montrer l'impact de l'état des moteurs, des expérimentations sont faites sur le robot Khepera 3 commandé par une station de contrôle avec une mission d'atteinte de cible. Le robot part d'une position initiale $(X_0, Y_0, \theta_0) = (0, 0, 0)$ pour arriver à une position finale $(1, 1)$. Le pourcentage d'erreur P_{erreur} a été défini dans l'équation 5.4. Lorsque $P_{erreur} \leq 10\%$, le comportement du robot est le même que dans le scénario de la figure 5.4. Quand $10\% < P_{erreur} \leq 30\%$, davantage d'oscillations sont observées au niveau de la vitesse angulaire ω entraînant une dégradation de la réponse de l'angle d'orientation θ du robot comme le montre la figure B.8. Le robot arrive à sa destination après 20s. Pour un pourcentage $P_{erreur} > 30\%$, ce qui implique un état *mauvais* du moteur, le robot arrive à sa destination mais dans un temps plus grand : 50s au lieu de 20s comme le montrent les courbes de $X(t)$, $Y(t)$ et $v(t)$ de la figure B.9. Cela implique une diminution du temps de réponse du système. Le comportement du robot de la figure B.9 montre une dégradation parce que le robot est encore instable au bout de 20s. En effet, il existe un dépassement dans la courbe de réponse de $X(t)$, tandis que les courbes de l'orientation du robot θ et sa vitesse angulaire ω montrent des oscillations tout au long de la mission avant qu'il ne se stabilise dans sa position finale après 50s. Dans le cas où l'état d'un moteur se dégrade jusqu'à être *mauvais* et que l'autre reste en *bon* état, le comportement du robot est *mauvais* de telle sorte que le temps de réponse du système est de 170s comme le montre la figure B.10. La courbe de réponse de $X(t)$ montre un dépassement important et celle

de $Y(t)$ présente des oscillations avant que le système ne se stabilise. Cette dégradation est visualisable sur la courbe de la trajectoire du robot où des oscillations sont également observées. Des oscillations (limite de stabilité) sur l'orientation du robot tout au long de sa trajectoire sont observées montrant le *mauvais* état du robot. En dépit de ces problèmes, le robot arrive à sa destination grâce au correcteur PI.

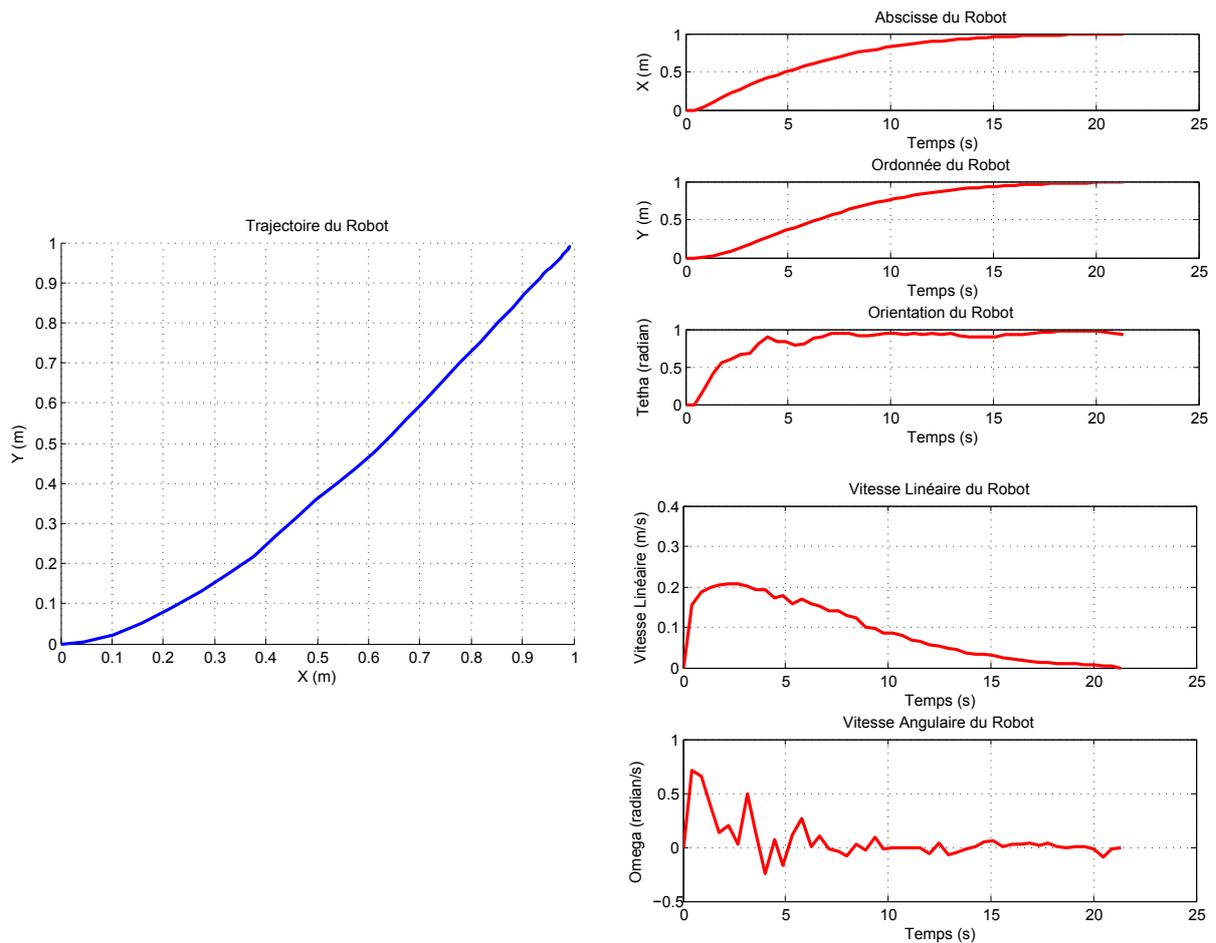
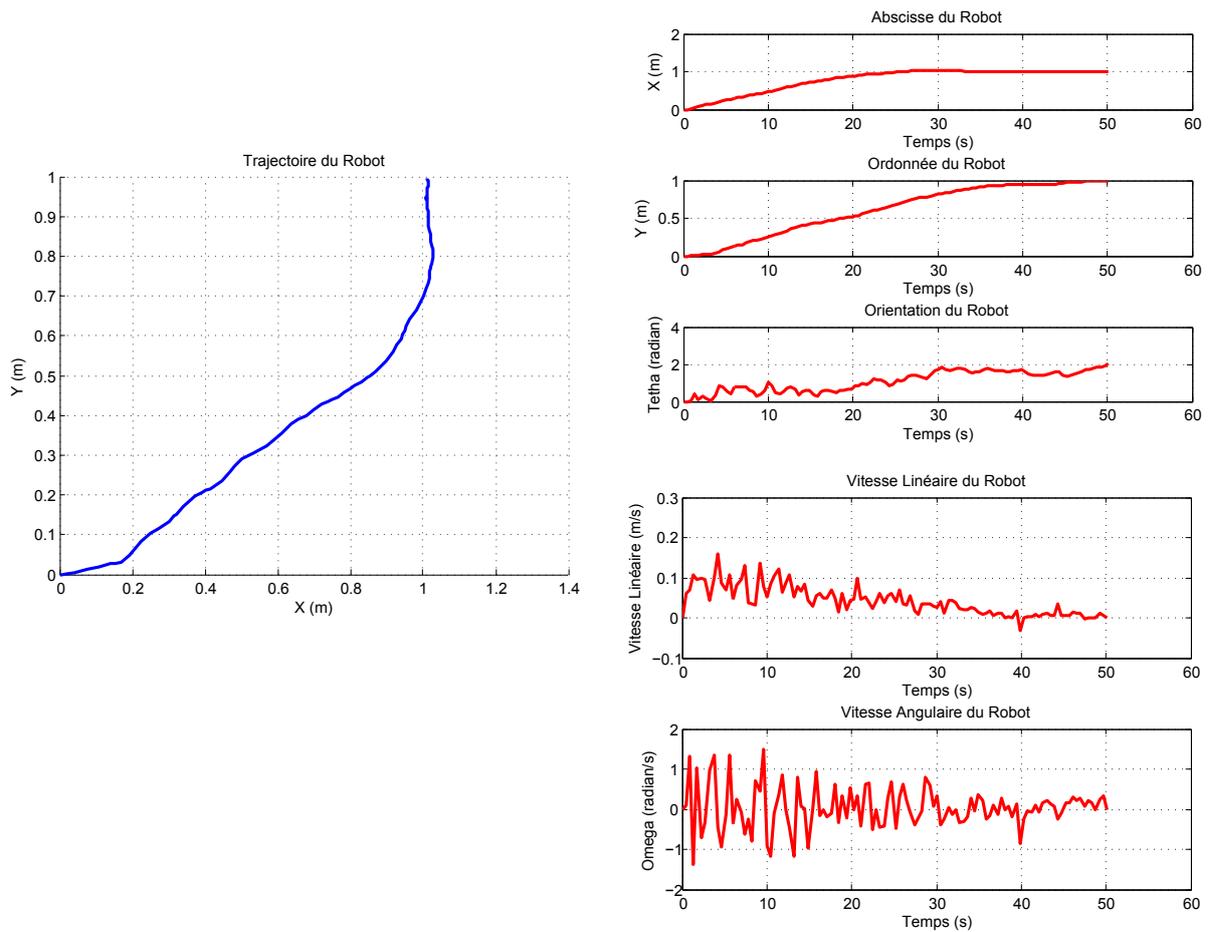


FIGURE B.8 – Comportement du robot dans le cas d'un état *dégradé* des moteurs

FIGURE B.9 – Comportement du robot dans le cas d'un état *mauvais* des moteurs

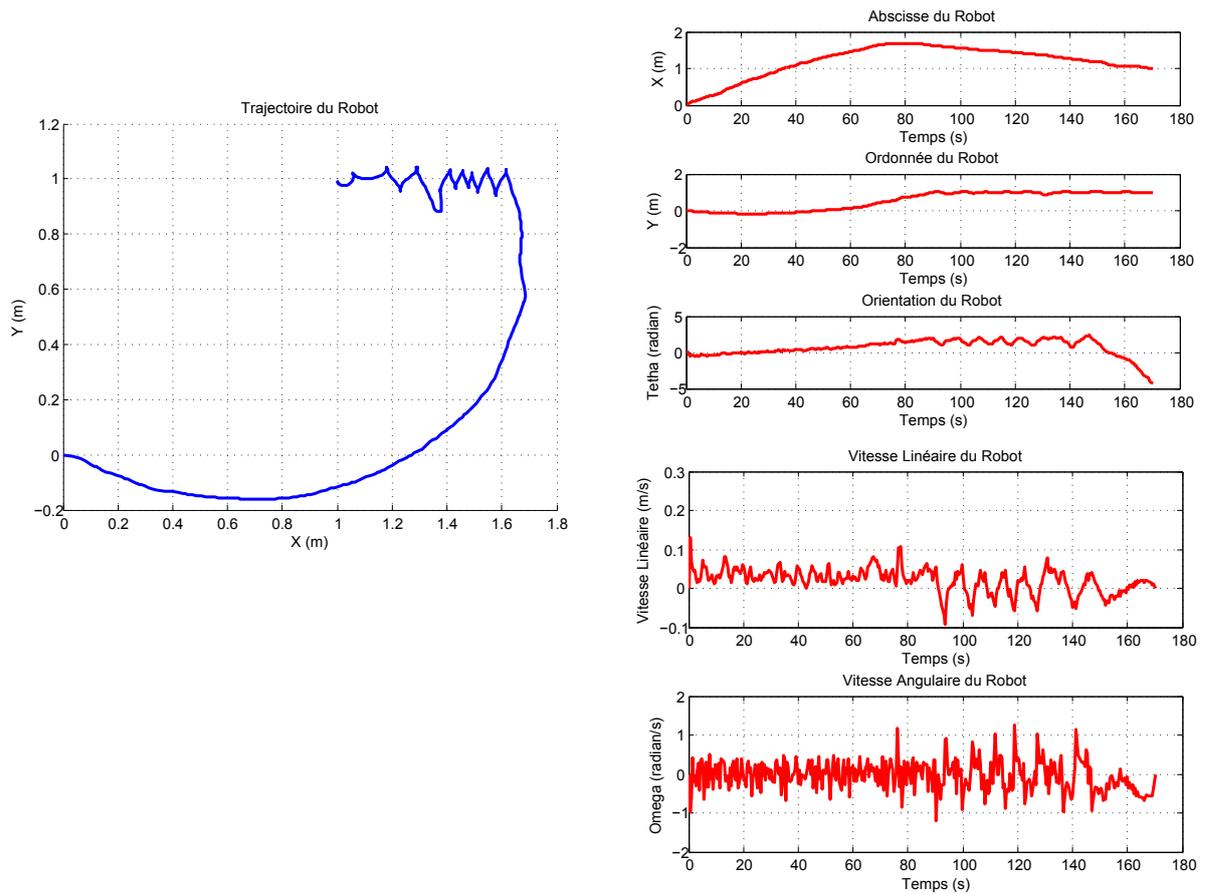


FIGURE B.10 – Comportement du robot dans le cas d'un moteur en *mauvais* état et d'un moteur en *bon* état

Bibliographie

- [ATC11] Pablo AROZARENA, Raquel TORIBIO et Alvaro CARRERA. « Distributed Fault Diagnosis Using Bayesian Reasoning in MAGNETO ». In : *Reliable Distributed Systems Workshops (SRDSW), 2011 30th IEEE Symposium on*. IEEE. 2011, p. 96–101 (cf. p. 52).
- [ÅW97] Karl J ÅSTRÖM et Björn WITTENMARK. « Computer controlled systems : Theory and design ». Prentice-Hall ». In : *Upper Saddle River NJ* (1997) (cf. p. 10, 11).
- [Ber+08] Cédric BERBRA et al. « Co-design of a safe network control quadrotor ». In : *Proceedings of 17th IFAC World Congress*. T. 8. 2008 (cf. p. 2).
- [Ber09] Cédric BERBRA. « Diagnostic des systèmes embarqués en réseau. Application à un mini drone hélicoptère ». Thèse de doct. Institut National Polytechnique de Grenoble-INPG, 2009 (cf. p. 23, 24).
- [Bia00] Giuseppe BIANCHI. « Performance analysis of the IEEE 802.11 distributed coordination function ». In : *IEEE Journal on selected areas in communications* 18.3 (2000), p. 535–547 (cf. p. 2, 20–22, 56, 57, 60, 68, 128).
- [Bot+08] Steven BOTTONE et al. « Failure prediction and diagnosis for satellite monitoring systems using Bayesian networks ». In : *Military Communications Conference, 2008. MILCOM 2008. IEEE*. IEEE. 2008, p. 1–7 (cf. p. 48).
- [BV02] Mark BLOEMEKE et Marco VALTORTA. *Agent-Encapsulated Bayesian Networks and The Rumor Problem*. Rapp. tech. Technical Reports, TR 2002-006, Depart of Computer Science, University of South Carolina, 2002 (cf. p. 39, 41).
- [CBV03] Periklis CHATZIMISIOS, Anthony C BOUCOUVALAS et Vasileios VITSAS. « IEEE 802.11 packet delay-a finite retry limit analysis ». In : *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*. T. 2. IEEE. 2003, p. 950–954 (cf. p. 69, 70).
- [Cen+07] Gianluca CENA et al. « Evaluation of response times in industrial WLANs ». In : *IEEE transactions on industrial Informatics* 3.3 (2007), p. 191–201 (cf. p. 16).
- [CHO10] Anton CERVIN, Dan HENRIKSSON et Martin OHLIN. « TRUETIME 2.0 beta—Reference Manual ». In : *Department of Automatic Control, Lund University (June 2010)* (2010) (cf. p. 81).
- [CIS07] J COLANDAIRAJ, GW IRWIN et WG SCANLON. « Wireless networked control systems with QoS-based sampling ». In : *IET Control Theory & Applications* 1.1 (2007), p. 430–438 (cf. p. 12, 15).

- [CP12] Jie CHEN et Ron J PATTON. *Robust model-based fault diagnosis for dynamic systems*. T. 3. Springer Science & Business Media, 2012 (cf. p. 47).
- [CXY13] Yu CHENG, Tianhua XU et Lianbao YANG. « Bayesian network based fault diagnosis and maintenance for high-speed train control systems ». In : *Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), 2013 International Conference on*. IEEE. 2013, p. 1753–1757 (cf. p. 49).
- [DC16] Ahmad W AL-DABBAGH et Tongwen CHEN. « Design Considerations for Wireless Networked Control Systems ». In : *IEEE Transactions on Industrial Electronics* 63.9 (2016), p. 5547–5557 (cf. p. 2, 15).
- [Dho03] Dominique DHOUTAUT. « Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation ». In : *Laboratoire CITI, INSA de Lyon* 11 (2003) (cf. p. 26).
- [DM03] José DEL SAGRADO et Serafin MORAL. « Qualitative combination of Bayesian networks ». In : *International Journal of Intelligent Systems* 18.2 (2003), p. 237–249 (cf. p. 116).
- [F+05] Chuan Heng FOH, Juki Wirawan TANTRA et al. « Comments on IEEE 802. 11 saturation throughput analysis with freezing of backoff counters ». In : *IEEE Communications Letters* 9.2 (2005), p. 130–132 (cf. p. 22).
- [Far11] Shahin FARAHANI. *ZigBee wireless networks and transceivers*. Newnes, 2011 (cf. p. 19).
- [Fer+13] Paolo FERRARI et al. « Improving simulation of wireless networked control systems based on WirelessHART ». In : *Computer Standards & Interfaces* 35.6 (2013), p. 605–615 (cf. p. 81).
- [FGG97] Nir FRIEDMAN, Dan GEIGER et Moises GOLDSZMIDT. « Bayesian network classifiers ». In : *Machine learning* 29.2-3 (1997), p. 131–163 (cf. p. 29).
- [Fra+14] Korbinian FRANK et al. « Segmenting Bayesian networks for intelligent information dissemination in collaborative, context-aware environments with Bayeslets ». In : *Pervasive and Mobile Computing* 12 (2014), p. 214–231 (cf. p. 34, 35).
- [GC08] Rachana A GUPTA et Mo-Yuen CHOW. « Overview of networked control systems ». In : *Networked Control Systems*. Springer, 2008, p. 1–23 (cf. p. 23).
- [GC10] Rachana Ashok GUPTA et Mo-Yuen CHOW. « Networked control system : Overview and research trends ». In : *IEEE transactions on industrial electronics* 57.7 (2010), p. 2527–2535 (cf. p. 13, 14, 23).

- [GFS12] Saeed GHAZANFARI-RAD, J-F FRIGON et Brunilde SANSÒ. « Theoretical framework for quality of service analysis of differentiated traffic in 802.11 wireless local area networks ». In : *IET communications* 6.15 (2012), p. 2326–2334 (cf. p. 66).
- [Gon+16] Apolinar GONZÁLEZ-POTES et al. « Distributed multi-agent architecture for real-time wireless control networks of multiple plants ». In : *Engineering Applications of Artificial Intelligence* 56 (2016), p. 142–156 (cf. p. 3, 23).
- [GTA11] Rony GHOSTINE, Jean-Marc THIRIET et Jean-François AUBRY. « Variable delays and message losses : influence on the reliability of a control loop ». In : *Reliability Engineering & System Safety* 96.1 (2011), p. 160–171 (cf. p. 23, 24).
- [Hab10] Gilbert HABIB. « Qualité de service et qualité de contrôle d’un Système Discret Contrôlé en Réseau Sans Fil : proposition d’une approche de co-conception appliquée au standard IEEE 802.11 ». Thèse de doct. Nancy 1, 2010 (cf. p. 8).
- [HC06] Keum-Sung HWANG et Sung-Bae CHO. « Modular Bayesian networks for inferring landmarks on mobile daily life ». In : *AI 2006 : Advances in Artificial Intelligence*. Springer, 2006, p. 929–933 (cf. p. 41).
- [HC09] Keum-Sung HWANG et Sung-Bae CHO. « Landmark detection from mobile life log using a modular Bayesian network model ». In : *Expert Systems with Applications* 36.10 (2009), p. 12065–12076 (cf. p. 42, 116).
- [HD96] Cecil HUANG et Adnan DARWICHE. « Inference in belief networks : A procedural guide ». In : *International journal of approximate reasoning* 15.3 (1996), p. 225–263 (cf. p. 32).
- [HNX07] Joao P HESPANHA, Payam NAGHSHTABRIZI et Yonggang XU. « A survey of recent results in networked control systems ». In : *Proceedings IEEE* 95.1 (2007), p. 138 (cf. p. 12, 23, 24).
- [HW13] Liangdong HU et Limin WANG. « Using consensus Bayesian network to model the reactive oxygen species regulatory pathway ». In : *PloS one* 8.2 (2013), e56832 (cf. p. 116, 117).
- [Ise05] Rolf ISERMANN. « Model-based fault-detection and diagnosis—status and applications ». In : *Annual Reviews in control* 29.1 (2005), p. 71–85 (cf. p. 48).
- [Jed09] Franck JEDRZEJEWSKI. « Chaînes de Markov ». In : *Modèles aléatoires et physique probabiliste* (2009), p. 67–88 (cf. p. 58–60).
- [KN03] Kevin B KORB et Ann E NICHOLSON. *Bayesian artificial intelligence*. T. 1. Florida : Chapman & Hall/CRC, 2003 (cf. p. 30, 32, 34).
- [Leg17] Lucas LEGON. « Mise en place d’une architecture de diagnostic pour un robot mobile en réseau ». In : *Rapport de stage d’initiation à la recherche* (2017) (cf. p. 92, 98, 142, 146, 147).

- [Li+15] Xiaomin LI et al. « A review of industrial wireless networks in the context of industry 4.0 ». In : *Wireless Networks* (2015), p. 1–19 (cf. p. 18).
- [LLC16] Sungsoo LIM, Seung-Hyun LEE et Sung-Bae CHO. « A modular approach to landmark detection based on a Bayesian network and categorized context logs ». In : *Information Sciences* 330 (2016), p. 145–156 (cf. p. 42–47, 52, 88, 128).
- [LM97] Kathryn Blackmond LASKEY et Suzanne M MAHONEY. « Network fragments : Representing knowledge for constructing probabilistic models ». In : *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1997, p. 334–341 (cf. p. 35).
- [LMT02] Feng-Li LIAN, James MOYNE et Dawn TILBURY. « Network design consideration for distributed control systems ». In : *Control Systems Technology, IEEE Transactions on* 10.2 (2002), p. 297–307 (cf. p. 10–12, 23).
- [Lu+12] Qiugang LU et al. « Analysis and synthesis for networked control systems with uncertain rate of packet losses ». In : *Journal of the Franklin Institute* 349.7 (2012), p. 2500–2514 (cf. p. 24).
- [Lup+14] Sergei LUPASHIN et al. « A platform for aerial robotics research and demonstration : The flying machine arena ». In : *Mechatronics* 24.1 (2014), p. 41–54 (cf. p. 3, 4).
- [LVB10] Scott LANGEVIN, Marco VALTORTA et Mark BLOEMEKE. « Agent-encapsulated bayesian networks and the rumor problem ». In : *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : volume 1-Volume 1*. International Foundation for Autonomous Agents et Multiagent Systems. 2010, p. 1553–1554 (cf. p. 39, 40).
- [LYC15] Seung-Hyun LEE, Kyon-Mo YANG et Sung-Bae CHO. « Integrated modular Bayesian networks with selective inference for context-aware decision making ». In : *Neurocomputing* 163 (2015), p. 38–46 (cf. p. 42, 43, 88, 99, 113, 128).
- [LZ05] Ioan Doré LANDAU et Gianluca ZITO. *Digital control systems : design, identification and implementation*. Springer Science & Business Media, 2005 (cf. p. 79).
- [Mal+15] Aqsa MALIK et al. « QoS in IEEE 802.11-based wireless networks : a contemporary review ». In : *Journal of Network and Computer Applications* 55 (2015), p. 24–46 (cf. p. 14).
- [Man+03] Stefan MANGOLD et al. « Analysis of IEEE 802.11 e for QoS support in wireless LANs ». In : *IEEE wireless communications* 10.6 (2003), p. 40–50 (cf. p. 14, 18).

- [MD14] Mark W MUELLER et Raffaello D'ANDREA. « Stability and control of a quadcopter despite the complete loss of one, two, or three propellers ». In : *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, p. 45–52 (cf. p. 4).
- [MDC13] Nicolas MARCHAND, Sylvain DURAND et Jose Fermi Guerrero CASTELLANOS. « A general formula for event-based stabilization of nonlinear systems ». In : *IEEE Transactions on Automatic Control* 58.5 (2013), p. 1332–1337 (cf. p. 10).
- [Mec10] Amine MECHRAOUI. « Co-conception d'un système commandé en réseau sans fil à l'aide de réseaux bayésiens distribués ». Thèse de doct. Institut National Polytechnique de Grenoble-INPG, 2010 (cf. p. 2, 8, 15, 24, 25, 79, 139, 144).
- [MKT09] Amine MECHRAOUI, Zeashan Hameed KHAN et Jean-Marc THIRIET. « Effect of packet loss on the quality of control of a networked mobile robot ». In : *30th IFAC Workshop on Real-Time Programming and 4th International Workshop on Real-Time Software*. 2009, p. 97–101 (cf. p. 24).
- [MMZ08] Kamal MEDJAHHER, Amine MECHRAOUI et Nouredine ZERHOUNI. « Diagnostic et pronostic de défaillances par réseaux bayésiens. » In : *4èmes Journées Francophones sur les Réseaux Bayésiens, JFRB'2008*. T. 1. 2008, p. 80–93 (cf. p. 47).
- [Mra+15] Ali Ben MRAD et al. « An explication of uncertain evidence in Bayesian networks : likelihood evidence and probabilistic evidence ». In : *Applied Intelligence* 43.4 (2015), p. 802–824 (cf. p. 33).
- [Mur+01] Kevin MURPHY et al. « The bayes net toolbox for matlab ». In : *Computing science and statistics* 33.2 (2001), p. 1024–1034 (cf. p. 107).
- [Nag+13] Keiji NAGATANI et al. « Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots ». In : *Journal of Field Robotics* 30.1 (2013), p. 44–63 (cf. p. 1).
- [Naï+07] P. NAÏM et al. *Les réseaux bayésiens*. 3^{ème} édition. Paris : Eyrolles, 2007 (cf. p. 28, 29, 32, 48).
- [NG04] Olfa Ben NACEUR-MOURALI et Christophe GONZALES. « Une unification des algorithmes d'inférence de Pearl et de Jensen. » In : *Revue d'Intelligence Artificielle* 18.2 (2004), p. 229–260 (cf. p. 32).
- [Oud+10] Patrick de OUDE et al. « Modular Bayesian networks : reasoning, verification and model inaccuracies ». In : *ASCI*. 2010 (cf. p. 41, 88, 113, 114).
- [PC07] K Wojtek PRZYTYLA et Arthur CHOI. « Reasoning framework for diagnosis and prognosis ». In : *Aerospace Conference, 2007 IEEE*. IEEE. 2007, p. 1–10 (cf. p. 30, 50–53, 89).

- [Pea88] Judea PEARL. *Probabilistic reasoning in intelligent systems : Networks of plausible reasoning*. 1988 (cf. p. 28, 31–33, 132, 133).
- [PG04] Mark A PASKIN et Carlos E GUESTRIN. « Robust probabilistic inference in distributed systems ». In : *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press. 2004, p. 436–445 (cf. p. 35).
- [PGM05] Mark PASKIN, Carlos GUESTRIN et Jim MCFADDEN. « A robust architecture for distributed inference in sensor networks ». In : *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press. 2005, p. 8 (cf. p. 35–37).
- [PH16] Chen PENG et Qing-Long HAN. « On designing a novel self-triggered sampling scheme for networked control systems with data losses and communication delays ». In : *IEEE Transactions on Industrial Electronics* 63.2 (2016), p. 1239–1248 (cf. p. 9).
- [Qu+16] Feng-Lin QU et al. « Quantized stabilization of wireless networked control systems with packet losses ». In : *ISA transactions* 64 (2016), p. 92–97 (cf. p. 24).
- [RKT17] Amy R. WAGONER, Daniel K. SCHRADER et Eric T. MATSON. « Towards a Vision-Based Targeting System for Counter Unmanned Aerial Systems (CUAS) ». In : *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA 2017)*. 2017 (cf. p. 4).
- [RMM11] Hala RIFAI, Samer MOHAMMED et Abdelhamid MELLOUK. « A brief synthesis of QoS-QoE methodologies ». In : *Programming and Systems (ISPS), 2011 10th International Symposium on*. IEEE. 2011, p. 32–38 (cf. p. 78).
- [RMT12] Nedjemi RACHEDI, Amine MECHRAOUI et Jean-Marc THIRIET. « Pronostic de défaillances d’un robot mobile commandé à travers un réseau sans fil ». In : *18ème congrès de maîtrise des risques et sûreté de fonctionnement (Lambda Mu 18)*. 2012, Communication–73 (cf. p. 101).
- [SGT15] Insaf SASSI, Alexia GOUIN et Jean-Marc THIRIET. « Diagnostic des systèmes mobiles communicants ». In : *Journées Nationales des Communications Terrestres 2015*. 2015 (cf. p. 3).
- [SGT16] Insaf SASSI, Alexia GOUIN et Jean-Marc THIRIET. « A Bayesian network for diagnosis of networked mobile robots ». In : *European Safety and Reliability Conference 2016*. 2016 (cf. p. 3).
- [SGT17a] Insaf SASSI, Alexia GOUIN et Jean-Marc THIRIET. « Diagnosis architecture reconfiguration for networked mobile robot ». In : *European Safety and Reliability Conference 2017*. 2017 (cf. p. 3).

- [SGT17b] Insaf SASSI, Alexia GOUIN et Jean-Marc THIRIET. « Distributed to Embedded Bayesian Network for Diagnosis of a Networked Robot ». In : *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA 2017)*. 2017 (cf. p. 3).
- [SGT17c] Insaf SASSI, Alexia GOUIN et Jean-Marc THIRIET. « Wireless Network Performance Evaluation for Networked Robots ». In : *Emerging Technologies and Factory Automation (ETFA), 2017 IEEE 22nd International Conference on*. IEEE. 2017 (cf. p. 2).
- [SK16] Angela SAINI et Tadafumi KOYAMA. « Cleanup technologies following Fukushima ». In : *MRS Bulletin* 41.12 (2016), p. 952–954 (cf. p. 1).
- [SL13] Mohamed Samir SAYED et Niels LOHSE. « Distributed Bayesian diagnosis for modular assembly systems—A case study ». In : *Journal of Manufacturing Systems* 32.3 (2013), p. 480–488 (cf. p. 52).
- [SMM12] Johann SCHUMANN, Timmy MBAYA et Ole J MENGSHOEL. « Software and system health management for autonomous robotics missions ». In : *Proc. of i-SAIRAS 2012* (2012) (cf. p. 49, 50).
- [SSA10] Daniel SIMON, Ye-Qiong SONG et Christophe AUBRUN. *Co-design approaches to dependable networked control systems*. John Wiley & Sons, 2010 (cf. p. 2, 8–10, 12–15, 23, 25).
- [Szt12] János SZTRIK. « Basic queueing theory ». In : *University of Debrecen, Faculty of Informatics* 193 (2012) (cf. p. 60, 67).
- [TBX10] Ilenia TINNIRELLO, Giuseppe BIANCHI et Yang XIAO. « Refinements on IEEE 802.11 distributed coordination function modeling approaches ». In : *IEEE Transactions on Vehicular Technology* 59.3 (2010), p. 1055–1067 (cf. p. 22).
- [TC03] Yodyium TIPSUWAN et Mo-Yuen CHOW. « Control methodologies in networked control systems ». In : *Control engineering practice* 11.10 (2003), p. 1099–1111 (cf. p. 8, 9, 14, 23, 24).
- [TCK07] Juan Marcos TOIBERO, R CARELLI et B KUCHEN. « Switching control of mobile robots for autonomous navigation in unknown environments ». In : *Robotics and Automation, 2007 IEEE International Conference on*. IEEE. 2007, p. 1974–1979 (cf. p. 79, 143, 144, 146, 147).
- [TCT16] Guosong TIAN, Seyit CAMTEPE et Yu-Chu TIAN. « A deadline-constrained 802.11 MAC Protocol with QoS differentiation for soft real-time control ». In : *IEEE Transactions on Industrial Informatics* 12.2 (2016), p. 544–554 (cf. p. 2, 9, 14, 57, 62, 66).
- [Toi+11] Juan Marcos TOIBERO et al. « Switching control approach for stable navigation of mobile robots in unknown environments ». In : *Robotics and Computer-Integrated Manufacturing* 27.3 (2011), p. 558–568 (cf. p. 79).

- [WP17] Philippe WEBER et Luigi PORTINALE. *Special section on : Applications of Probabilistic Graphical Models in Dependability, Diagnosis and Prognosis*. 2017 (cf. p. 28, 48).
- [WS13] Philippe WEBER et Christophe SIMON. *Réseaux bayésiens : un nouveau formalisme de modélisation pour la sûreté de fonctionnement*. 2013 (cf. p. 48).
- [XG99] Y XIANG et H GENG. « Distributed equipment monitoring and diagnosis with multiply sectioned bayesian networks ». In : *AAAI spring symposium on AI in equipment service maintenance and support*. 1999, p. 18–25 (cf. p. 51, 52).
- [Xia00] Yang XIANG. « Belief updating in multiply sectioned Bayesian networks without repeated local propagations ». In : *International Journal of Approximate Reasoning* 23.1 (2000), p. 1–21 (cf. p. 38, 39, 41).
- [XL00] Y. XIANG et V. LESSER. « Justifying multiply sectioned Bayesian networks ». In : *Proceedings Fourth International Conference*. Sous la dir. d'IEEE. IEEE, 2000. Chap. MultiAgent Systems, p. 349–356 (cf. p. 38).
- [XL03] Yang XIANG et Victor LESSER. « On the role of multiply sectioned Bayesian networks to cooperative multiagent systems ». In : *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans* 33.4 (2003), p. 489–501 (cf. p. 114).
- [Xu+17] D XU et al. « Fault-tolerant control for wireless networked control systems with an integrated scheduler ». In : *International Journal of Adaptive Control and Signal Processing* 31.1 (2017), p. 23–38 (cf. p. 2, 15).
- [ZA02] Eustathia ZIOUVA et Theodore ANTONAKOPOULOS. « CSMA/CA performance under high traffic conditions : throughput and delay analysis ». In : *Computer communications* 25.3 (2002), p. 313–321 (cf. p. 22).
- [ZGK13] Lixian ZHANG, Huijun GAO et Okyay KAYNAK. « Network-induced constraints in networked control systems—a survey ». In : *Industrial Informatics, IEEE Transactions on* 9.1 (2013), p. 403–416 (cf. p. 23).
- [Zha+08] Liqiang ZHAO et al. « Integrated quality-of-service differentiation over IEEE 802.11 wireless LANs ». In : *IET communications* 2.2 (2008), p. 329–335 (cf. p. 14).
- [ZHY16] Xian-Ming ZHANG, Qing-Long HAN et Xinghuo YU. « Survey on recent advances in networked control systems ». In : *IEEE Transactions on Industrial Informatics* 12.5 (2016), p. 1740–1752 (cf. p. 10, 11, 23, 24).

Résumé — Dans la robotique mobile, le réseau de communication est un composant important du système global pour que le système accomplisse sa mission. Dans un tel type de système, appelé un système commandé en réseau sans fil (SCR sans fil ou WNCS), l'intégration du réseau sans fil dans la boucle de commande introduit des problèmes qui ont un impact sur la performance et la stabilité i.e, sur la qualité de commande (QoC). Cette QoC dépend alors de la qualité de service (QoS) et la performance du système va donc dépendre des paramètres de la QoS. C'est ainsi que l'étude de l'influence des défauts du réseau sans fil sur la QoC est cruciale. Le WNCS est un système temps réel qui a besoin d'un certain niveau de QoS pour une bonne performance. Cependant, la nature probabiliste du protocole de communication CSMA/CA utilisé dans la plupart des technologies sans fil ne garantit pas les contraintes temps réel. Il faut alors une méthode probabiliste pour analyser et définir les exigences de l'application en termes de QoS, c'est-à-dire en termes de délai, de gigue, de débit, et de perte de paquets. Une première contribution de cette thèse consiste à étudier les performances et la fiabilité d'un réseau sans fil IEEE 802.11 pour des WNCSs qui partagent le même réseau et le même serveur de commandes en développant un modèle stochastique. Ce modèle est une chaîne de Markov qui modélise la méthode d'accès au canal de communication. Ce modèle a servi pour définir les paramètres de la QoS qui peuvent garantir une bonne QoC. Nous appliquons notre approche à un robot mobile commandé par une station distante. Le robot mobile a pour mission d'atteindre une cible. Pour garantir l'accomplissement de cette mission, une méthode de diagnostic probabiliste est primordiale puisque le comportement du système n'est pas déterministe. La deuxième contribution a été d'établir la méthode probabiliste qui sert à surveiller le bon déroulement de la mission et l'état du robot. C'est un réseau bayésien (RB) modulaire qui modélise les relations de dépendance cause-à-effet entre les défaillances qui ont un impact sur la QoC du système. La dégradation de la QoC peut être due soit à un problème lié à l'état interne du robot, soit à un problème lié à la QoS, soit à un problème lié au contrôleur lui-même. Les résultats du modèle markovien sont utilisés dans le RB modulaire pour définir l'espace d'état de ses variables (étude qualitative) et pour définir les probabilités conditionnelles de l'état de la QoS (étude quantitative). Le RB permet d'éviter la dégradation de la QoC en prenant la bonne décision qui assure la continuité de la mission. En effet, dans une approche de co-design, quand le RB détecte une dégradation de la QoC due à une mauvaise QoS, la station envoie un ordre au robot pour qu'il change son mode de fonctionnement ou qu'il commute sur un autre contrôleur débarqué. Notre hypothèse est que l'architecture de diagnostic est différente en fonction des modes de fonctionnement : nous optons pour un RB plus global et partagé lorsque le robot est connecté à la station et pour RB interne au robot lorsqu'il est autonome. La commutation d'un mode de fonctionnement débarqué à un mode embarqué implique la mise à jour du RB. Un autre apport de cette thèse est

la définition d'une stratégie de commutation entre les modes de diagnostic : commutation d'un RB distribué à un RB monolithique embarqué quand le réseau de communication ne fait plus partie de l'architecture du système et vice-versa. Les résultats d'inférence et de scénario de diagnostic ont montré la pertinence de l'utilisation des RBs distribués modulaires. Ils ont aussi montré la capacité du RB développé à détecter la dégradation de la QoC et de la QoS et à superviser l'état du robot. L'aspect modulaire du RB a permis de faciliter la reconfiguration de l'outil de diagnostic selon l'architecture de commande ou de communication adaptée (RB distribué ou RB monolithique embarqué).

Mots clés : diagnostic, réseau bayésiens, réseau de communication, systèmes en réseau, sûreté de fonctionnement.

Abstract — In mobile robotics systems, the communication network is an important component of the overall system, it enables the system to accomplish its mission. Such a system is called Wireless Networked Control System WNCS where the integration of the wireless network into the control loop introduces problems that impact its performance and stability i.e, its quality of control (QoC). This QoC depends on the quality of service (QoS) therefore, the performance of the system depends on the parameters of the QoS. The study of the influence of wireless network defects on the QoC is crucial. WNCS is considered as a real-time system that requires a certain level of QoS for good performance. However, the probabilistic behavior of the CSMA / CA communication protocol used in most wireless technologies does not guarantee real-time constraints. A probabilistic method is then needed to analyze and define the application requirements in terms of QoS : delay, jitter, rate, packet loss. A first contribution of this thesis is to study the performance and reliability of an IEEE 802.11 wireless network for WNCSs that share the same network and the same control server by developing a stochastic model. This model is a Markov chain that models the access procedure to the communication channel. This model is used to define the QoS parameters that can guarantee the good QoC. In this thesis, we apply our approach to a mobile robot controlled by a remote station. The mobile robot aims to reach a target, a classic example of mobile robotics applications. To ensure that its mission is accomplished, a probabilistic diagnostic method is essential because the system behavior is not deterministic. The second contribution of this thesis is to establish the probabilistic method used to monitor the robot mission and state. It is a modular Bayesian network BN that models cause-and-effect dependency relationships between failures that have an impact on the system QoC. The QoC degradation may be due either to a problem related to the internal state of the robot, a QoS problem or a controller problem. The results of the Markov model analysis are used in the modular BN to define its variables states (qualitative

study) and to define the conditional probabilities of the QoS (quantitative study). It is an approach that permits to avoid the QoC degradation by making the right decision that ensures the continuity of the mission. In a co-design approach, when the BN detects a degradation of the QoC due to a bad QoS, the station sends an order to the robot to change its operation mode or to switch to another distant controller. Our hypothesis is that the diagnostic architecture depends on the operation mode. A distributed BN is used when the robot is connected to the station and a monolithic embedded BN when it is autonomous. Switching from a distributed controller to an on-board one involves updating the developed BN. Another contribution of this thesis consists in defining a switching strategy between the diagnostic modes : switching from a distributed BN to an on-board monolithic BN when the communication network takes no longer part of the system architecture and vice versa -versa. The inference and diagnostic scenarii results show the relevance of using distributed modular BNs. They also prove the ability of the developed BN to detect the degradation of QoC and QoS and to supervise the state of the robot. The modular structure of the BN facilitates the reconfiguration of the diagnostic policy according to the adapted control and communication architecture (distributed BN or on-board monolithic RB).

Keywords : diagnostic, bayesian network, communication networks, networked systems, dependability

GIPSA-lab, 11 rue des Mathématiques
38402 Saint Martin d'Hères